## Project Plan

# Remote Exploration and Experimentation (REE) Project

July 6, 2000

Agreements:

| | |
|---|---|
| Edward C. Stone<br>Director<br>NASA Jet Propulsion Laboratory | Date |
| Eugene L. Tu<br>HPCC Program Manager<br>NASA Ames Research Center | Date |
| Robert D. Ferraro<br>Project Manager<br>NASA Jet Propulsion Laboratory | Date |

This online version of the REE Project Plan is for public reference. The current complete approved plan is available from the REE Project Manager for internal NASA use only.

Concurred by:

| | |
|---|---|
| Satish Khanna<br>Manager (Acting)<br>NASA Technology Program<br>Jet Propulsion Laboratory | Date |
| Michael J. Sander<br>Director<br>Technology and Applications Programs<br>Jet Propulsion Laboratory | Date |
| Joseph Bredekamp<br>HPCC Senior Program Executive<br>NASA Office of Space Science | Date |

This online version of the REE Project Plan is for public reference. The current complete approved plan is available from the REE Project Manager for internal NASA use only.

## TABLE OF CONTENTS

This online version of the REE Project Plan is for public reference. The current complete approved plan is available from the REE Project Manager for internal NASA use only.

## FORWARD

This document contains the Project Plan for the NASA Remote Exploration and Experimentation (REE) Project. This document is updated as required and is the controlling document that defines the technical and management structure of the Project. The Project described in this document will accelerate the development of high performance computing technologies to meet the needs of the spaceborne research community. It will also accelerate the distribution of these technologies to the American public. The technologies developed under this plan will help maintain U.S. technical and economic leadership in the international arena of high-performance computing. The time period covered by this plan is fiscal years 2000-2005.

The REE Project is a component of the NASA High Performance Computing and Communications (HPCC) Program, which in turn is part of the Federal program in Computing, Information and Communications (CIC). The primary goal of the Federal CIC effort is to extend U.S. technological leadership in high performance computing and computer communications. As this is accomplished, these technologies will be widely disseminated to accelerate the pace of innovation and improve national economic competitiveness, national security, education, health care, and the global environment. The NASA HPCC program is a critical element of the Federal CIC effort.

NASA's primary contribution to the Federal program is its leadership in the development of algorithms and software for high-end computing and communication systems which will increase system effectiveness and reliability, as well as support the deployment of high-performance, interoperable, and portable computational tools. As HPCC technologies are developed, NASA will use them to address aerospace transportation systems, Earth sciences, and space sciences research challenges. NASA's specific research challenges include improving the design and operation of advanced aerospace transportation systems, increasing scientists' abilities to model the Earth's climate and predict global environmental trends, further our understanding of our cosmic origins and destiny, and improving the capabilities of advanced spacecraft to explore the Earth and solar system. The HPCC Program supports research, development, and prototyping of technology and tools for education, with a focus on making NASA's data and knowledge accessible to America's students. These challenges require significant increases in computational power, network speed, and the system software required to make these resources effective in real-world science and engineering environments.

HPCC is a research program that pursues computing and communications technologies at various levels of maturity. It is structured to contribute to the broad Federal CIC effort, while addressing agency-specific computational problems that are beyond projected near-term computing capabilities. Computational problems in the areas of Earth science, space science, and aerospace are used as drivers of this research, providing the context and requirements for the work that is to be done. This work—and the HPCC Program—is organized into three high-end computing projects, a high-performance communications project, and an education project.

- Computational Aerospace Sciences (CAS)
- Earth and Space Sciences (ESS)
- Remote Exploration and Experimentation (REE)
- NASA Research and Education Network (NREN)

- Learning Technologies (LT)

These Projects, and their associated applications, were chosen for their potential and direct impact to NASA, their national importance, and the technical challenge they give the NASA HPCC Program. The document describing this program is the *High Performance Computing and Communications (HPCC) Program Plan.*

This online version of the REE Project Plan is for public reference. The current complete approved plan is available from the REE Project Manager for internal NASA use only.

# 1 Introduction

The Remote Exploration and Experimentation (REE) Project is one of five Projects in the High Performance Computing and Communications (HPCC) Program. The Program is governed by the HPCC Program Commitment Agreement. Begun in 1992 as one of the original three Projects in the HPCC Program, it was deferred from 1993 – 1996 due to budget constraints. The Jet Propulsion Laboratory is the Lead Center for the REE Project. At this time, the Goddard Space Flight Center is supporting the REE Project by providing two science application teams and characterizing the radiation effects exhibited by commercial computing technology components.

## 1.1 HISTORY

> *"The REE element addresses critical needs to both the Offices of Space Science and Mission to Planet Earth. A new generation of on-board computers will enhance science return, reduce operations costs, and mitigate downlink limitations"* [1]
>
> *Wesley T. Huntress, Jr.*
> *Associate Administrator for Space Science*
>
> *Charles F. Kennel*
> *Associate Administrator for Mission to Planet Earth*

It was with these prescient words in mind that the Workshop on Remote Exploration and Experimentation was convened in Pasadena, CA on August 21–23, 1995. The Workshop was followed by a study phase that took place in fiscal years 1996 and 1997. During this period, the REE Project consulted with US leaders in spaceborne avionics, high-performance computing, commercial computing manufacturers, other government agencies, and NASA Space and Earth Scientists to devise a strategy and approach for meeting its then scheduled Program Commitment Agreement (PCA) milestone in September 2003*: Demonstration of spaceborne applications on embedded high performance computing testbed.* Key technical issues were examined, including: the current state-of-the-art in spaceborne embedded computing systems, the trends in technology development for both spaceborne and commercial ground-based computing systems, and the projected computing requirements for several classes of NASA missions in the next millennium. Based on the results of the study phase, the REE Project developed a Vision and a set of Goals and Objectives that define the Project and its expected outcome. From these Goals and Objectives, a schedule of Project Milestones was developed which led to demonstration of NASA spaceborne applications on a high performance embedded computing system in space.

In 1999, the HPCC Program undertook a reassessment of its goals and objectives in response to a request by the HPCC Program Executive Committee. The Program examined its alignment with the changing needs of the NASA Enterprises that it serves, and made adjustments to its commitments based upon this assessment. The PCA and Program Plan were updated to reflect these adjustments and to emphasize the cross-enterprise and cross-project nature of its activities.

[1] *Letter to: R/Director, High Performance Computing and Communications Office, July 6, 1994*

In particular, the PCA milestone *Demonstration of spaceborne applications on embedded high performance computing testbed* was eliminated from the PCA in favor of a Program level milestone of the same character. This Project Plan reflects those adjustments. The vision, goals, and objectives of the Remote Exploration and Experimentation Project were found to remain relevant and timely with respect to Enterprise needs. Some adjustments to the Project's schedule were made so that it more accurately reflects the current realities of technology advancements and progress against original milestones. A complete list of changes to the Project may be found in the change log at the end of this document.

## 1.2 VISION AND GOALS

The commercial computing industry is two orders of magnitude larger than the entire space and defense electronics industry, and each year this disparity grows larger. The government no longer is a driving force in the state-of-the-art development of computing technology, and has little influence over its direction. At the same time, NASA and DOD requirements for space-capable computing technology are becoming more demanding, especially with regard to available power and cooling, performance, reliability, and cost. The REE Project seeks to leverage the considerable investment by the ground-based computing industry to bring supercomputing technologies into space within the constraints imposed by that environment. The availability of onboard computing capability will enable a new way of doing science in space at significantly reduced overall cost. The vision of the REE Project, therefore, is:

> *To bring commercial supercomputing technology into space, in a form which meets the demanding environmental requirements, to enable a new class of science investigation and discovery.*

Derived from this vision, REE has identified two principal goals. Specifically, the REE Project will:

> Demonstrate a process for rapidly transferring commercial high performance computing technology into low power, fault tolerant architectures for space.
>
> Demonstrate that high performance onboard processing capability enables a new class of science investigation and highly autonomous remote operation.

The legacy of the REE Project will not only be a new generation of scalable onboard supercomputing in space, but the validation of a process which will keep spaceborne computing capabilities on the same technology track as the commercial computing industry.

## 1.3 OVERALL APPROACH/TIMEFRAME

Based on the results of the Study Phase, the REE Project developed a Technology and Applications Roadmap that leads to the attainment of the Project's goals and objectives. This roadmap is shown in Figure 1. It consists of three parallel interdependent initiatives, supporting the development of computing testbeds, system software, and applications. In addition, there is a system engineering effort that assures the overarching coordination of these three initiatives. These initiatives—guided by the system engineering team—work in concert with each other to achieve the demonstration of scalable spaceborne applications on a high performance embedded scalable computing testbed.

**Computing Testbeds Initiative.** The purpose of the Computing Testbeds initiative is to explore and develop a process for translating commercial high performance scalable parallel computing architectures into low power spaceborne implementations. These architectures must rely, to the maximum extent practical, on commercial-off-the-shelf (COTS) technologies and must minimize or eliminate the use of radiation-hardened components. The process must be consistent with the rapid (18 months or less) transfer of new earth-based technologies to NASA space missions. Translated architectures must satisfy a number of additional criteria, including no single point of failure and graceful performance degradation in the event of hardware failure.

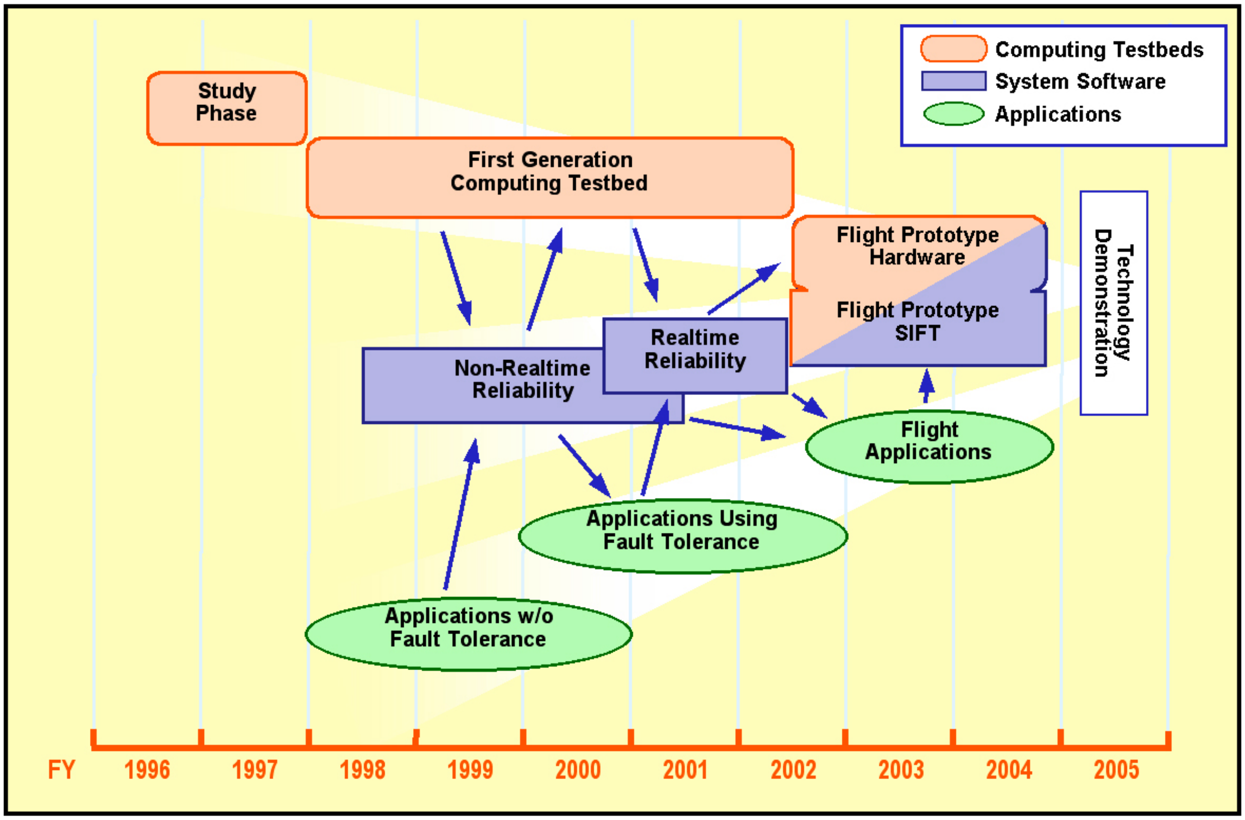

**Figure 1. Technology and Applications Roadmap for the REE Project. This roadmap calls for the parallel development of hardware testbeds, systems software, and applications.**

The Computing Testbeds initiative will develop a series of hardware prototypes, leading to the demonstration of a capability of at least *300 MOPS*[2]*/watt.* This represents an increase of two orders of magnitude over the power performance of the flight computer onboard the Mars Pathfinder spacecraft that landed on Mars in July 1997. At the present time, a hardware testbed, called the First Generation Testbed (FGT), is being developed to demonstrate that significant power performance *(30 MOPS/watt)* can be achieved in a scalable embedded architecture using commercial technology. This testbed will also be the platform for conducting software implemented fault tolerance experiments and for developing the system software needed to

*[2] MOPS: Millions of Operations Per Second. These may be a mixture of 32 bit integer and floating point arithmetic or logical operations. Although MIPS (Millions of Instructions per Second) is a more traditional measure of processor capability, it does not quantify the actual amount of work accomplished on processors which have complex instruction sets. In many cases, however, MOPS and MIPS will be interchangeable*

achieve the reliability goals. The FGT will be delivered to JPL in June 2000. Two years following this delivery, the REE Project will begin development of a prototype flight computer. The architecture of this platform will be based on the experience gained with the FGT. In addition, it will match the mass and form factor of a future flight model and will demonstrate scalability *(50 nodes),* reliability *(0.99 over five years),* and a power performance of at least *300 MOPS/watt.* The prototype flight computer will be delivered to JPL in March 2004.

**System Software Initiative.** The purpose of the System Software initiative is to provide a set of services that will enable applications to take full advantage of the computing capacity of the hardware architecture, while providing an easy-to-use development environment and assuring reliable operation in space. By relying to the maximum extent practical on commercial software components, the system software layer will provide for the requisite performance capability and user interface. However, no commercially available parallel processing system offers a significant level of fault tolerance without substantial task replication. Since the hardware architecture will be based on commercially available components, radiation-induced faults will be common and hardware component failure will be a possibility. Hence, the system software must provide mechanisms for recovery from both permanent and transient faults. It will be a major challenge to the System Software initiative to develop a fault detection and recovery scheme that assures system reliability without compromising the performance capability available to the applications.

The System Software initiative will develop a middleware layer between a commercial operating system and the applications. This middleware layer will offer a suite of fault tolerance mechanisms from which the applications can make selections based on their reliability and efficiency requirements. The first version of the middleware layer will demonstrate reliability based on software implemented fault tolerance *(0.99 over five years),* scalability *(50 nodes),* and portability for all REE applications. A later revision will add real-time capability as a feature.

**Applications Initiative.** The purpose of the Applications initiative is to demonstrate that the unique high performance low-power computing capability developed by the Project enables new science investigation and discovery. Science Application teams will demonstrate that substantial onboard computational capability will be a crucial ingredient in science investigations of the future. They will ensure that architectures and system software produced by the Project meet the needs of the spaceborne applications community. They will stimulate the development and implementation of new computational techniques that will transform the REE platforms from computers into tools of scientific discovery, on a par with the sensors and data collection systems with which they are integrated.

The Applications teams will develop scalable science and autonomy application algorithms. Software will be developed and installed on the Computing Testbeds hardware. This software will be used to test, evaluate, and validate candidate architectures and system software using the REE testbed. A demonstration of scalable applications on the First Generation Testbed will take place within months of its delivery. Subsequent generations of scalable applications for installation on the REE flight computer prototype will build on the experience gained on the previous Computing Testbeds hardware. These applications will be demonstrated on the flight prototype. Although not a requirement for successful completion of the Project, REE will actively seek an opportunity to demonstrate the flight prototype in space. This opportunity would be in the form of an engineering demonstration of capability, with the mission execution costs (launch, operations, etc.) being borne by another program, such as the New Millennium Program.

**System Engineering.** The purpose of the System Engineering effort is to define and document the detailed requirements of the REE System and to integrate and test the outputs of the Applications, Computing Testbeds, and System Software initiatives. These requirements will be developed in preparation for major procurements and will address both hardware architecture and system software requirements. As part of this effort, the System Engineering Team will develop radiation fault models that are applicable at both the component and system levels. These models will help establish total integrated dose (TID) tolerances and orbit-dependent fault models for all system components. In addition, the System Engineering Team will define the overall architecture of the REE System and will conduct a Critical Design Review (CDR) of the flight prototype system design. It will also test and validate all project deliverables to assure that the delivered products conform to the system requirements.

***Raison d’être.*** These parallel tracks for the development of technology and applications are of equal importance. The significance of this point cannot be overemphasized. It is in the delivery of more science at lower cost that REE finds its ultimate *raison d’être.* This has motivated both an involvement in Space Science Enterprise and Earth Science Enterprise long-term planning and the creation of the Applications branch on the REE roadmap. It is through the involvement of users that the Project will introduce scalable spaceborne computing to the space science and autonomy communities and unearth the new mission concepts enabled by REE.

## 2 Objectives

From the Project Vision and Goals, REE has developed four specific Objectives:

1. Demonstrate power efficiencies of at least 300 MOPS per watt in an architecture that can be scaled up to 100 watts, depending on mission needs.
2. Demonstrate new spaceborne applications on embedded high performance computing testbeds, which return analysis results to the Earth in addition to raw data.
3. Develop fault-tolerant designs that will permit reliable operation for 5 years and more using commercially available or derived components.
4. Investigate ultra-low power (> 1000 MOPS/watt) onboard computer systems which will help open the entire Solar System to exploration without the need for nuclear technology.

These objectives address key issues in response to spaceborne computing requirements for the future. From the HPCC Program heritage of scalable multiprocessor systems, REE has derived its reliance on the commercial computing investment to provide components and architectures that have the capability to address NASA’s onboard computing needs. The translation of HPCC technology to space, however, requires the Project to address issues of power, fault tolerance, and reliability which are different from the concerns of ground based computing. In particular, the limited available onboard power, the lack of ability to repair or replace failed components, and the need to compute in an environment which produces transient faults define the 1st, 3rd, and 4th objectives of the Project. The second objective is prompted by a history of robotic space science missions which did nothing more than compress the data collected before transmitting it to the ground. The REE Project intends to demonstrate the usefulness of high performance embedded computing technology for enhancing the science returned in the presence of limited bandwidth to the ground and restrictive communications latencies to the spacecraft.

These objectives have driven the planning of the Project Milestones, which define the path to be taken towards the achievement of these objectives. Each milestone has a set of output metrics which define the required capability in hardware, applications performance, software reliability, and overall system performance. These metrics are defined later in this document along with the Project Milestones in the **SCHEDULES** section.

# 3 Customer Definition and Advocacy

A fundamental goal of the REE Project is to enable the return to Earth of dramatically new science results and insight from NASA spacecraft, using the unique high performance low-power spaceborne computing capability developed by the Project. REE is a technology push project, designed to inject the HPCC Program scalable computing technology into NASA’s spaceborne exploration activities. The customer base REE seeks to satisfy is future NASA science missions that face severe constraints on onboard power, cost, and communications bandwidth and latency to the ground. REE technology will not initially be targeted for used in routine spacecraft control (e.g., attitude control and thruster firing), although there is nothing inherent in our approach that precludes this. Such control tasks are not compute-bound because they are designed to be managed by state-of-the-art single-string radiation-hardened processors. REE technology will initially be used to provide high throughput processing (with high availability) for data-prolific science instruments. It may in addition be used for “spacecraft control” in the sense that for some applications, onboard computing will enable real-time redirection of the observing program based on the identification of a science target of opportunity. The Project is chartered to take the risk in introducing the latest commercial technology into space, solve the reliability and implementation problems, and transfer the technology to the mainstream of NASA’s space missions. In order to adopt this new technology, the mission customers must be convinced that their reliability is not compromised, their capability is enhanced, and their budgets are not negatively impacted.

To achieve this goal, REE has engaged high profile mission scientists to lead its applications teams. It is the science mission principle investigator who will ultimately define the required science return, which in turn sets the requirements for spacecraft capability. The Project seeks to maximize that return for a given cost by enabling new scientific investigations supported by capable onboard computing. These investigations will be defined by our primary customers, the science research community, and in particular by space and Earth science instrument Principal Investigators.

Currently, five teams of science and autonomy investigators have been assembled by the REE Project to put forth specific proposals for novel applications to exploit the scalable hardware and system software. These teams are listed in Table 1. They are performing the following crucial functions:

1) Developing revolutionary new mission concepts that utilize substantial onboard computational power as a crucial ingredient in scientific data collection, analysis, editing, and discovery.
2) Ensuring that architectures and system software produced under the Project match the scientific needs of the spaceborne applications community.
3) Driving the implementation of new algorithms and computational techniques that transform the REE platforms from computing devices to tools of scientific discovery, on a par with the sensors and data collection devices with which they are integrated.
4) Forming the nucleus of an extended community of advocates for the utilization of spaceborne computing as a tool for remote exploration and experimentation in the planning and execution of NASA missions.

**Table 1. REE Science Application Teams**

| Application | Principal Investigator | NASA Theme Addressed |
|---|---|---|
| Gamma Ray Large Area Space Telescope (GLAST) | Prof. Thompson Burnett<br>University of Washington | Structure and Evolution of the Universe |
| Mars Rover Science | Dr. Steven Saunders<br>Jet Propulsion Laboratory | Exploration of the Solar System |
| Next Generation Space Telescope (NGST) | Dr. John Mather<br>Goddard Space Flight Center | Search for Origins |
| Orbiting Thermal Imaging Spectrometer | Prof. Alan Gillespie<br>University of Washington | Earth Science Enterprise |
| Solar Terrestrial Probe | Dr. Steven Curtis<br>Goddard Space Flight Center | Sun-Earth Connection |

Throughout the life of the Project, the set of science application teams will evolve to continue to cover mission areas that are of importance to NASA. It is the advocacy of these missions which is crucial to the Project’s success.

In addition to science customers, REE must also meet the requirements of mission engineers who must integrate this technology into the next generation of spacecraft. To ensure that the technology developed by the Project will be compatible with future missions, REE will engage in discussions with advanced mission avionics and mission data systems developers to determine interoperability and compatibility requirements to which the flight prototype will adhere in order that it be both flight-ready and mission-insertable in the 2005 time frame.

# 4 Project Authority

The overall project authority for the REE Project is established by the HPCC Program, which is in turn established by the NASA Headquarters Program Management Council. The HPCC Program Commitment Agreement (PCA) represents the Agency-level agreement for the implementation of the HPCC Program and its Projects. Although the program is funded by three Enterprises and the NASA Office of Human Resources and Education, the overall management of HPCC is formally within the Aerospace Enterprise and is the responsibility of the HPCC Program Office at the NASA Ames Research Center (ARC).

The Jet Propulsion Laboratory is the designated lead center for the Remote Exploration and Experimentation Project. JPL has REE Project Management authority and responsibility. The NASA Goddard Space Flight Center (GSFC) supports the REE Project through its participation in the development of algorithms and software for the Next Generation Space Telescope and the Solar Terrestrial Probe science application teams, and in characterizing the radiation tolerance of commercial computing technology components. Other NASA centers may be called upon from time to time to support specific development activities in the Project as needed.

# 5 Management

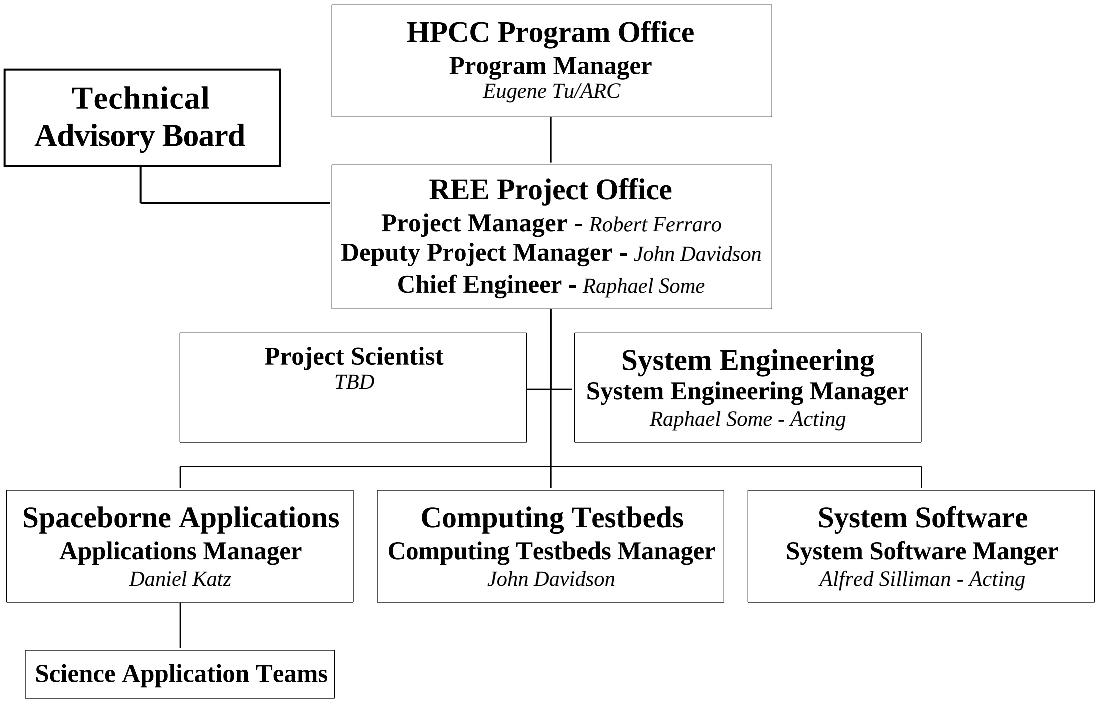

**Figure 2. Management structure of the REE Project**

## 5.1 ORGANIZATION

The REE Project is managed by the REE Project Manager who reports to JPL Center Management and to the HPCC Program Manager. The REE Project Manager directs and controls the day-to-day activities necessary to accomplish Project goals and ensure customer satisfaction. Work performed at JPL for this project will be executed according to Policies and Procedures of the JPL Develop Needed Technology Domain. The REE Project Manager is assisted by a Deputy Project Manager, a Chief Engineer, and a Project Scientist. The major Work Breakdown Structure (WBS) elements are each lead by an element Manager who is responsible for the day-to-day activities within these areas. A suite of Applications have been identified and put under contract to the REE Project. Each Application has designated a Principal Investigator, who oversees day-to-day activities and reports to the REE Applications Manager. Figure 2 shows the management structure of the Remote Exploration and Experimentation Project.

### 5.1.1 Technical Advisory Board

In Fiscal Year 2000, REE will form a Technical Advisory Board. The purpose of the Technical Advisory Board is to periodically review the progress and plans of the Project for consistency, feasibility, and compatibility with spacecraft architecture constraints. This Board will be composed of recognized experts from academia, NASA stakeholders, and other interested Federal agencies. The Board will meet at least once a year to review the Project and advise Project Management of any corrective actions that should be taken to assure success. It may meet more frequently as the need arises.

## 5.2 RESPONSIBILITIES

### 5.2.1 Project Manager

The overall management of the Remote Exploration and Experimentation Project is the responsibility of the REE Project Manager who is appointed from the Technology and Applications Programs Directorate at JPL. The specific responsibilities of the REE Project Manager are:

(a) Develop, update, and maintain the REE Project Plan, including the definition and negotiation of resource, schedule, and deliverable commitments, in cooperation with functional managers in the participating and sponsoring organizations.

(b) Direct and control the day-to-day activities necessary to accomplish the goals and objectives of the Project and to ensure customer satisfaction.

(c) Coordinate REE activities with those of the other HPCC Projects and participate through the HPCC Program in the Federal Program in Computing, Information and Communications.

(d) Coordinate REE activities with those of related programs in other government agencies, such as the Defense Advanced Research Projects Agency (DARPA) and the Air Force Research Labs.

(e) Appoint the Deputy Project Manager, the Chief Engineer, Project Scientist, and Managers for Applications, Computing Testbeds, System Software, and System Engineering, and define and interpret element area responsibilities.

(f) Approve the Project Implementation Plan.

(g) Achieve all Project milestones.

### 5.2.2 Deputy Project Manager

The Deputy Project Manager is appointed by the Project Manager and assists the Project Manager in the development of the Project Plan, reporting and review activities, and the day-to-day operation of the project.

### 5.2.3 Chief Engineer

The Chief Engineer is appointed by the Project Manager. The specific responsibilities of the Chief Engineer are:

(a) Prepare and maintain the Project Implementation Plan, which specifies the requirements, task level milestones and integrated schedule, and subsidiary controlled documents.

(b) Define the overall system architecture, approaches to achieving Project Milestones, and methods for measuring deliverables against output metrics.

(c) Organize, coordinate and direct the technical activities of the Project. Approve element task milestones and schedules.

(d) Review and approve all technical documentation including the documentation tree, task plans, test procedures, specifications, requirements, and reports.

### 5.2.4 Project Scientist

The Project Scientist is appointed by the Project Manager. The specific responsibilities of the Project Scientist are:

(a) Represent the interests of the REE Application Teams to the Project.

(b) Stand in for other NASA Science Investigators (both Space and Earth Science) in representing their interest to the Project.

(c) Promote the REE Project interests and accomplishments at science venues at NASA Headquarters and on other occasions as they arise.

(d) Participate in Project planning and reviews to assure that the future needs of NASA science flight missions are represented.

### 5.2.5 Applications Manager

The Applications Manager is appointed by the Project Manager. The specific responsibilities of the Applications Manager are:

(a) Define the task milestones and schedules necessary to achieve the Applications Milestones.

(b) Manage the cost, schedule, procurements and technical activities for the Applications element.

(c) Provide technical assistance to the Application Teams as needed.

(d) Represent the Application Teams requirements to the Project and to the computing testbeds, system software, and system engineering activities.

(e) Provide the applications necessary to achieve all Project milestones.

### 5.2.6 Computing Testbeds Manager

The Computing Testbeds Manager is appointed by the Project Manager.  The specific responsibilities of the Computing Testbeds Manager are:

(a) Define the task milestones and schedules necessary to achieve the Computing Testbeds Milestones.

(b) Manage the cost, schedule, procurements and technical activities for the Computing Testbeds element.

(c) Provide the computing testbeds infrastructure necessary to achieve all Project milestones.

### 5.2.7 System Software Manager

The System Software Manager is appointed by the Project Manager.  The specific responsibilities of the System Software Manager are:

(a) Define the task milestones and schedules necessary to achieve the System Software Milestones.

(b) Manage the cost, schedule, procurements and technical activities for the System Software element.

(c) Oversee the design, implementation, and testing of software implemented fault tolerance layers.

(d) Provide the software implemented fault tolerance necessary to achieve all Project milestones.

### 5.2.8 System Engineering Manager

The System Engineering Manager is appointed by the Project Manager.  The specific responsibilities of the System Engineering Manager are:

(a) Define the task milestones and schedules necessary to achieve the System Engineering Milestones.

(b) Manage the cost, schedule, procurements and technical activities for the System Engineering element.

(c) With the support of the other element mangers, define all project test procedures, and integrate and test all project deliverables.

(d) Conduct system design studies and define the system level fault model.  Conduct fault and risk management studies.

### 5.2.9 Field Center Responsibilities

The Jet Propulsion Laboratory is the lead center for the REE Project.  JPL will provide the technical lead and Project Management for REE.  The Goddard Space Flight Center is a support center to the REE Project. GSFC provides science application teams to the Project, and assists in the radiation performance characterization of commercial technologies.

### 5.2.10 Reporting Responsibilities

The REE Project Manager will submit status, management, and financial reports to the HPCC Program Manager as specified in the HPCC Program Plan. On an annual basis, the REE Project Manager will prepare an accomplishments summary suitable for inclusion in the HPCC Annual Report.

### 5.2.11 Coordination with Related Programs

The REE Project will coordinate its activities with those of related programs in other government agencies. In particular, REE is closely coordinating its activities with Air Force Research Laboratory (AFRL) Improved Space Computer Program (ISCP). REE will also coordinate its activities with other NASA next generation space avionics and mission software initiatives to ensure compatibility, interoperability and insert-ability into future spacecraft and missions.

## 6 Technical Summary

The Technical Summary is divided into five major subsections. These are: Applications, Computing Testbeds, System Software, System Engineering, and Advanced Technology Opportunities. The relationships among the first three major Project activities and the strategy behind their structure was shown in Figure 1. Advanced Technology Opportunities are high risk, high payoff investments which potentially crosscut the first three activities and could result in REE significantly exceeding its Project goals, should they be successful. However, these investments are not on the Project's critical path. The purpose of the System Engineering activity is to provide an overarching integration and coordination function for the technical activities of the project elements and to provide for the testing and validation of all project deliverables.

The lifetime of the Project can be divided into four somewhat overlapping phases. These are a study phase, a testbed development phase, an experiment phase, and a flight prototype phase.

In the study phase, it was determined that the Project was feasible and quantitative goals were defined. In this phase, the current state-of-the-art in spaceborne embedded computing systems was examined. Trends in technology development and the projected computing requirements for several classes of NASA missions were assessed. Based on the results of this phase, a set of project objectives was developed, including the objective of demonstrating a power efficiency of 30 MOPS per watt by 2000 and 300 MOPS per watt by 2004. This performance would be demonstrated in an architecture that could be scaled up to 100 watts, depending on mission needs. The study phase took place and was completed during fiscal years 1996–1997.

In the testbed development phase, the REE First Generation Testbed (FGT) is being built under a contract with Sanders, a Lockheed Martin Company. This contract calls for Sanders to deliver a testbed consisting of twenty fully functioning hardware nodes. The most important specifications for this testbed are that it deliver a power performance of 30 MOPS per watt and provide fault injection capabilities for simulating the space environment. In addition, a small testbed, called the Level Zero Testbed, has been assembled at JPL out of commercial parts and is currently providing a low-cost interim environment for the development of application and system software prior to the delivery of the First Generation Testbed. Although the Level Zero Testbed does not attain the power performance and fault tolerance of the FGT, it replicates its interfaces and functionality in most essential aspects. The testbed development phase is taking place during fiscal years 1998–2001.

In the experiment phase, the Level Zero Testbed and the First Generation Testbed are used to explore systems concepts which utilize software as well as hardware to achieve reliability. To achieve fault tolerance and reliability in a COTS-based architecture, the REE Project will take a systems-level approach. Experiments in software implemented fault tolerance (SIFT), using NASA applications as benchmarks, will be performed to understand how system-level reliability can be achieved without the need for radiation-hardening of individual components. Additional science teams will be engaged to expand the range of applications and broaden the new science thrust of the project. The experiment phase will take place during fiscal years 2000–2002.

In the flight prototype phase, the lessons learned from the experiment phase will be used to create a protoflight system that is form, fit and function flight-ready. In this phase, a flight ready system with state-of-the-art hardware and software components in an optimized architectural configuration will be fabricated and demonstrated. The final system will be tested, validated in a laboratory setting, and qualified for flight. Although a space-based demonstration is not called for in the project plan, it is anticipated that there will be flight opportunities available for such a demonstration, beginning in fiscal year 2004. The flight prototype phase will take place during fiscal years 2002–2004.

The final result of these four phases will be a flight system demonstration and, potentially, a flight experiment in the 2004–2005 time frame. Such an experiment, while not required in a programmatic sense, remains a goal of the REE Project as demonstration of the potential of a spaceborne supercomputer for enabling a new class of science investigation and discovery. Thus, in a larger sense, the result of these four phases will be an architecture and an approach that will enhance science return, reduce operations costs, and revolutionize the way that scientific research is done in space.

The following sections detail each of the major activities that contribute to these four phases of the Project.

## 6.1 APPLICATIONS

A fundamental goal of REE is to enable the return to Earth of dramatically new scientific results and insight from NASA spacecraft, using the unique high performance low-power spaceborne computing capability developed by the Project. To achieve this goal, new scientific directions will be defined by the scientific research community, especially by space and Earth science instrument Principal Investigators.

### 6.1.1 Science Strategy and Approach

Five teams of science and autonomy investigators have been assembled by the REE Project to put forth specific proposals for novel science applications to exploit the scalable hardware and system software. These teams perform the following crucial functions:

1) Develop revolutionary new mission concepts that utilize substantial onboard computational power as a crucial ingredient in scientific data collection, analysis, editing, and discovery.
2) Ensure that architectures and system software produced under the Project match the scientific needs of the spaceborne applications community.
3) Drive the implementation of new algorithms and computational techniques that transform the REE platforms from computing devices to tools of scientific discovery, on a par with the sensors and data collection devices with which they are integrated.

The REE Applications highlight entirely new ideas. There are two fundamental reasons for this. First, they have available the unique resources supplied by REE: at least two orders of magnitude more computational power than has previously been available in space. Second, these resources may be deployed on miniature spacecraft orders of magnitude smaller than those currently in existence, with severely limited electrical power for data transmission to earth.

Spacecraft autonomy is already a vigorous focus of future spacecraft planning at NASA. It is a major goal, for example, in NASA's New Millennium Program. The unique ingredient that will be provided by REE is the ability to pursue science-driven autonomy, which is currently considered only in research programs such as the Office of Space Science Autonomy and Operations Technology Program. For example, an REE computer would enable vigilant spacecraft, or spacecraft fleets, that will be able to monitor planetary, Earth, solar and stellar targets continuously for weeks, months, even years at a time. Applications implemented on the REE computer will be able to flag hazardous or scientifically interesting events as they occur, allowing the spacecraft to respond autonomously, either to maintain its own health in the face of hazards, or to image especially interesting behavior at higher resolution so that the most scientifically important results can be returned to Earth.

Note that the onboard computing capability provided by REE is absolutely crucial to the achievement of scientific goals in situations in which a rapid adaptive response to unexpected events is needed. For example, it may be required to capitalize on important transient activity in an imaged target. This point is often greatly under-appreciated. It is typically assumed, for example, that the decision to rely upon substantial onboard computing to process scientific data depends solely upon the telemetry bandwidth available to transmit this data to Earth. However, in regimes such as deep space, there is often insufficient time to return data to Earth and to await further instructions during an interesting unexpected transient occurrence, *even if sufficient telemetry bandwidth is available.* In other cases, future competition for resources such as NASA's Deep Space Network will severely limit the amount of downlink available to individual missions, even for spacecraft operating in relatively power-rich environments such as an orbit of Venus.

Accomplishment of science-driven autonomy goals will require a suite of new algorithms and applications software to be developed as part of REE, to ensure that hardware capabilities of the REE computers are exploited to their fullest. These include automated onboard data analysis of remote sensing imagery, autonomous navigation and control software, planning and scheduling of resources, data compression and editing, and the construction of onboard catalogues and models as scientific reference points in the knowledge discovery process. These activities must be defined and prioritized by the science community as part of their involvement with REE.

Each Application team has the following Project responsibilities:

1) Identify important new scientific directions that may be enabled by REE.
2) Analyze the computational requirements, especially with respect to CPU, RAM, I/O, sophistication of programming model, importance of fault tolerance, and operating system needs. These analyses, which were completed in the first six months of the current science teams' contracts, are used to evaluate testbed architectures.
3) Develop algorithms and prototype applications on ground testbeds to demonstrate feasibility. Initially, access was provided to traditional HPC platforms such as the Cray T3E, SGI Origin, and HP Exemplar machines, supporting parallel APIs such as MPI. These have been augmented by ground testbeds, as described in the Computing Testbeds sub-section below.
4) Explore new approaches to science onboard in a limited downlink bandwidth /high downlink latency environment.
5) Assist in fault-behavior experiments and in development of application-based fault detection and handling techniques.

The initial Applications teams have become an integral part of the Project, and they, as well as future application teams, are ultimately responsible for the applications required in the Project's milestones as outlined in Section 7.

### 6.1.2 Selection of Application Teams

Five application teams were selected in fiscal years 1997/1998 to participate in the REE Project. They are: Gamma Ray Large Area Space Telescope (GLAST), Next Generation Space Telescope (NGST), Mars Rover Science, Orbiting Thermal Imaging Spectrometer (OTIS), and Solar Terrestrial Probe Multiplatform Missions. These teams are led by NASA scientists. They are developing algorithms and software for applications that emphasize *in-situ* analysis of science instrument data and remote operation of highly autonomous systems. These applications were chosen on the basis of their potential for benefiting from the hundred-fold increase in onboard computing power that REE promises. The operation of their instruments is constrained because of the combination of their high data rates or limitations in spacecraft downlink bandwidth, or both. In some cases, operation is constrained by latency.

These current Application Teams, their science objectives, and the attributes that drive their science requirement are described in detail in **Appendix D**. The applications developed by these teams will be used over the life of the Project. In order to maintain a strong connection to current NASA mission directions, the set of Application Teams will be periodically refreshed. In particular, an Application Team with a real time processing requirement will be selected during Fiscal Year 2000.

The REE Project anticipates a competitive solicitation for the addition of new application teams starting in FY01. The purpose of the new application teams will be to add to the diversity of the

REE application set, specifically in terms of real-time vs. non-real-time applications, autonomous applications, amount of required computation, balance between parallel and distributed processing, and dynamic requirements for resources. A peer review process will be used to evaluate proposed application teams, including reviewers from government, academia, and industry. Each proposed team will be required to have a funded tie to a NASA flight project, to ensure the relevance of the teams to NASA Enterprises.

### 6.1.3 Application Libraries

The REE application teams have requested a complete set of applications libraries, providing functionality in linear algebra, signal and image processing, and statistics. An effort is underway to create robust parallel versions of these libraries, using Algorithm-Based Fault tolerance (ABFT) and Result-Checking techniques originally developed in academia. New versions of parallel linear algebra routines and fast-Fourier Transform routines have already been demonstrated which can detect errors in computation with very low increase in computational overhead. Other libraries and techniques will be developed as needed to modify the applications required to demonstrate fulfillment of the System Software milestones.

## 6.2 COMPUTING TESTBEDS

The purpose of the Computing Testbeds activity is to transition commercial scalable high performance computing architectures into forms that are appropriate for a spaceborne computer. This spaceborne computer must rely, to the maximum extent practical, on commercial-off-the-shelf technologies and must minimize or eliminate the use of radiation-hardened components. The approach must be consistent with the rapid (18 months or less) transfer of new earth-based technologies to NASA space missions. The architectures must satisfy a number of additional criteria, including no single point of failure and graceful performance degradation in the event of component failure.

The Computing Testbeds initiative consists of three distinct phases. The first of these was a study phase. This was successfully executed and completed, establishing the feasibility of the Project's goals and objectives. The second phase entails the development of a hardware testbed that will demonstrate scalability *(50 nodes)* and power performance *(30 MOPS/watt).* This testbed is called the First Generation Testbed (FGT). The FGT completed its design phase in September 1998. It is being fabricated and will be delivered in June 2000. The third phase of the Computing Testbeds initiative calls for the development of a flight prototype. This prototype will match the mass and form factor of a future flight model and will demonstrate scalability *(50 nodes),* reliability *(0.99 over five years),* and a power performance of at least *300 MOPS/watt.* This represents an increase of two orders of magnitude over the state-of-the-art.[3] The hardware prototype will be delivered to JPL in June 2004.

In the following subsections, the First Generation Testbed and the Flight Prototype are discussed in detail.

[3] *Mars Pathfinder, July, 1997*

### 6.2.1 First Generation Scalable Embedded Computing Testbed

Beginning in fiscal year 1997, the REE Project formed a collaborative relationship with industry to develop a first-generation scalable, high performance, low-power computing testbed. This testbed will be used to demonstrate scalability *(50 nodes)* and system-level power performance (at least *30 MOPS per watt).* It will be used to test, refine, and validate scalable system approaches to fault tolerance prior to investing in the development of a flight prototype by providing fault injection capabilities which mimic the space radiation environment. This testbed will be delivered to JPL in June 2000 and upgraded in 2001.

The development of the First Generation Testbed has proceeded as follows. A solicitation was issued at the end of fiscal year 1997 inviting proposals from teams led by industry, or possibly academia, to develop a testbed platform to investigate scalable low-power high performance architectures, based largely on COTS technologies. Following evaluation of the proposals received, contracts were awarded to two teams, led by: Sanders, a Lockheed-Martin Company, of Nashua, NH and SEAKR Engineering, Inc. of Englewood, CO. (Table 3) These vendors were selected for a six-month Design Phase, at the end of which one would be selected to fabricate and deliver the hardware testbed. The testbed design is required to contain 20 nodes, at least four of which are fully functional hardware nodes, capable of demonstrating the power performance requirement. Although this testbed will contain only 20 nodes, it can be used to investigate scalability to larger configurations by a combination of experiment and analysis. Applications developed by the Applications Teams will be ported to the testbed to evaluate the total system performance for a variety of spaceborne computing scenarios.

**Table 3. Participants in the REE Testbed Design Phase (3/98-9/98).**

| Lead Organization | Collaborating Organizations |
|---|---|
| Sanders, a Lockheed-Martin Company | California Institute of Technology<br>Lockheed Martin Federal Systems<br>University of Illinois<br>University of Southern California |
| SEAKR Engineering, Inc. | Lockheed Martin Control Systems<br>Lockheed Martin Tactical Defense Systems<br>Motorola Corp.<br>SGI/Cray Research, Inc. |

At the end of the Design Phase, Design Reviews were held at the home facilities of the two lead organizations. Based on an evaluation of these proposals by a source evaluation team, Sanders was selected to fabricate and deliver the First Generation Testbed. Sanders was placed on contract with JPL in early November 1998. This contract calls for Sanders to deliver a testbed consisting of twenty fully functioning hardware nodes, running the Lynx real-time operating system, communicating over a message passing interconnect supplied by Myricom Corp. Fault injection and fault monitoring software will be developed at Sanders and delivered with the testbed. The most important specification for this testbed is that it deliver a power performance of 30 MOPS per watt. In Sanders' architecture, the key to attaining this power performance is a special purpose ASIC that manages internode communication. Sanders calls this ASIC the "Node Controller." (In its initial implementation, the Node Controller will be implemented through the development of an FPGA.) The First Generation Testbed will be delivered in June 2000.

In addition, a small testbed, called the Level Zero Testbed, has been assembled at JPL out of commercial parts and is currently providing a low-cost interim environment for the development of application and system software prior to the delivery of the First Generation Testbed. Although the Level Zero Testbed does not attain the power performance and fault tolerance of the FGT, it replicates its interfaces and functionality in most essential aspects. The Level Zero Testbed was initially established in March 1998 to permit software development to move forward while waiting for the selection of a vendor to build the First Generation Testbed.

The REE application teams have requested a set of parallel programming tools (e.g., performance monitors and debuggers) to assist in their code development. Commercially available parallel programming tools will be provided as part of the testbed delivery. In addition, the REE Project will coordinate with the CAS and ESS Projects to determine the applicability of the tools developed under these Projects to the REE environment.

### 6.2.2 Flight Prototype Embedded Scalable Computer

In fiscal year 2002, the REE Project will begin development of a prototype flight computer. In this phase, a flight ready system with state-of-the-art hardware and software components in an optimized architectural configuration is fabricated and demonstrated. The architecture of this platform will be based on the refinements developed through the experience with the FGT. The flight prototype will match the mass and form factor of a future flight model and will demonstrate scalability *(50 nodes)*, reliability *(0.99 over five years)*, and a power performance of at least *300 MOPS/watt.* As with the FGT, the prototype will be developed in partnership with industry. Additional science teams will be engaged to expand the range of applications and broaden the new-science thrust of the project. Application software developed by the Applications Teams will be installed on the flight prototype. The prototype will be used to demonstrate a low power, scalable architecture capability using the latest generation COTS and low power component technologies with a systems level approach to fault tolerance and real-time capability. Although a space-based demonstration of the prototype is not required for successful completion of the Project, it is anticipated that there will be several flight opportunities available for such a demonstration in the *2004 – 2005* timeframe. The Flight Prototype Computer will be delivered to JPL in March 2004.

## 6.3 SYSTEM SOFTWARE

The primary goal of the REE system software effort is to provide a set of services which enables applications to take full advantage of the computing capacity of the REE architecture while providing an easy-to-use programming and development environment. In addition, the system software must provide for fault detection and fault recovery so that applications can operate in the presence of faults. To the maximum extent possible, a commercial scalable multiprocessor operating system will be baselined and the added functionality will be layered on top of it.

The System Software activity consists of research and development efforts that explore the capabilities and limitations of software solutions to the fault tolerance problem that results from using non-radiation hardened COTS technology in space. Radiation in the space environment will induce random transient errors in these components at rates that vary with position of the spacecraft and solar activity. These errors can result in corruption of the result of computation or of system state. Traditional fault tolerance approaches handle the problem through hardware architecture and the use of radiation hardened components that minimize these transient errors. REE seeks to take advantage of the substantial speed advantage state-of-the-art non-radhard COTS components have over radiation hardened components by implementing these fault tolerance techniques in software. However, there will exist error rates above which a software solution is not feasible. The critical task of the system software activity will be to explore a variety of techniques with varying overheads and reliabilities which still provide an overall system level advantage over traditional hardware approaches. It must also determine the fault rate limit at which this approach no longer makes sense.

In this context, it is important to distinguish between distributed applications and parallel applications. Both sets of applications are spread across multiple processors. Distributed applications are characterized by several cooperating tasks with multiple tasks per processor. These tasks are loosely coupled in the sense that they communicate infrequently, and the communication protocols that they employ can consist of several protocol layers without impacting the overall performance of the system. Parallel applications, by contrast, are characterized by several cooperating tasks in which a single task may be spread across multiple processors. The processors communicate frequently, and the communication protocols must be extremely efficient in order not to impact the overall performance of the system. Indeed, distributed systems may copy messages several times as messages are passed from one protocol layer to another, while parallel systems go to great lengths to avoid copying messages even once.

The REE Project is focused primarily on applications that are parallel. To date, most of the work in fault-tolerant multiprocessors has focused on distributed applications where fault tolerance can be implemented via relatively expensive mechanisms such as message duplication and task replication and voting, and relatively little attention has been paid to parallel applications. (Indeed, no commercially available parallel processing system offers any significant level of fault tolerance.) The challenge for the REE Project is to develop a system that provides fault tolerance with as little overhead as possible based on the reliability requirements of the application.

Preliminary experiments during fiscal years 1998 and 1999 resulted in the demonstration of software techniques to detect errors in certain types of computations and to initiate automatic recovery from these errors. ABFT techniques with low overhead were defined for certain classes of linear algebra computations and demonstrated to be effective in detecting errors. In partnership with the University of Illinois, a process monitoring system call Chameleon was refined and demonstrated at error rates that were an order of magnitude higher than is expected of current generation COTS components in low Earth orbit or in deep space. These experiments validated the notion that a software approach to fault tolerance was feasible for at least some class of onboard processing applications. The immediate challenge is to extend this work to address the full spectrum of onboard processing applications and system software, and to understand the limits of its applicability. This software implemented fault tolerance (SIFT) will ultimately consist of a set of techniques in a middleware layer which augments the normal embedded system OS and applications.

The near term objective of the SIFT development activity is to demonstration an initial capability that will provide high system reliability (*0.99 over five years*) and high system availability (*0.99 over 5 years*). In addition, SIFT must support scalability and applications portability. A variety of techniques will be developed, tested, and assessed for their limits of applicability. This capability will be demonstrated using REE applications on the hardware testbed by March 2001. The next objective is to add real time processing capability *(50 ms performance latencies)* for certain kinds of applications to the system. This added capability will be demonstrated by March 2002. Once this period of development and experimentation is concluded in 2002, a complete SIFT layer redesign will be undertaken to integrate the lessons learn. A prototype of the redesigned SIFT layer will be tested and demonstrated in 2003, and a final integrated system will be delivered in 2004.

### 6.3.1 User Access

A fundamental requirement of the REE-based system is that it be easy to use and that it support the needs of the user community. Ideally, the user will develop, validate, and update application software on his or her laboratory workstation. Updated software will then be installed on the REE platform and operated with the same user interface as on the workstation. To facilitate this goal, the system software will utilize tools, interfaces, and programming languages that are based on standards and commercial products that are familiar to the user community.

The REE hardware system will consist of a set of computing nodes and memory interconnected by a network fabric. Based on input from the REE Applications teams, it appears that a relatively simple programming model (based on explicit message passing) will be sufficient for REE applications. In addition, the system software must provide task management to enable task creation, deletion, context switching, and scheduling. Also, the operating system on each node will provide memory protection facilities which will “fence off” applications from the operating system and each other, allowing multiple applications to run on any given node. The system software must also provide access to mass storage with an appropriate I/O model. The size of the prototype REE applications is relatively small, so it is not anticipated that paging or swapping functions will be required. A system software layer built from commercial software components should be able to provide these functions. In addition, the REE Project may investigate parallel languages and alternate programming models as candidate technologies for testbed validation.

### 6.3.2 Fault Tolerance and Real Time Operation

Real time operation is defined as the ability to respond to an external event, such as an externally generated signal to the system, and take appropriate action within a specified period of time. The time period must take into account the complexity of the response, but a guarantee of action within a defined period is what distinguishes real time systems from non-real time systems. The REE Project has set as its target a latency of no more than 50 milliseconds between the time an externally generated signal is input to the systems and the time at which the first instruction of the signal handler is executed. This latency target is to be met even in the presence of faults occurring at rates characteristic of low Earth orbit. An underlying assumption is that applications which require real time operation can be executed successfully on a single node of an REE system.

The REE Project will approach the development of software-implemented fault tolerance and real-time capability primarily through partnering with industry. We will look to the private sector for cooperative development of a SIFT middleware layer to provide reliable operation on high performance parallel hardware. It is essential that the hardware and software be developed concurrently so that meaningful tradeoffs can be made during the design of both, resulting in the optimum system design.

## 6.4 SYSTEM ENGINEERING

The purpose of the System Engineering effort is to define system requirements and define the overall architecture of the REE system. The System Engineering effort is also responsible for the integration of the Project's technical activities and for the testing and validation of all project deliverables.

### 6.4.1 System Definition

*System Requirements.* The System Engineering effort will define and document the detailed requirements of the REE System. These requirements will be developed in preparation for major procurements and will address both hardware architecture and system software requirements. They will be derived from the goals, objectives, milestones, and output metrics as stated in this project plan. For example, it is assumed that the system will be based on COTS technology. In addition, it is assumed that new technology will be moved from the commercial market into spaceborne use within 18 months. On an architectural level, it is assumed that the system contains processing nodes linked by a high-speed network, that this system is extremely rich in connectivity and in processing resources, and that it will be scaled to meet a range of mission computing needs and mass/power constraints. In the area of fault tolerance, the requirements will be based on the generally accepted guidelines for NASA flight missions. For example, the REE system shall have an availability of 99% and a reliability of 99% over 5 years.

It is crucial to the development of system requirements that fault models be developed that are applicable at both the component and system levels. How often will the system be disrupted by a cosmic ray-induced single event upset (SEU)? How will these SEUs be distributed throughout the system? The System Engineering effort will attack these questions in several ways. A suite of data types will be combined to produce a radiation fault model, including engineering data from space- and accelerator-based experiments. In parallel with the collection of these data, computer-based models will be developed and validated. Together, these efforts will serve to establish total integrated dose (TID) tolerances and orbit-dependent fault models for all system components. Based on these rates, fault injection experiments will be conducted to determine response at the system level to the effects of cosmic radiation and to validate the effectiveness of various SIFT approaches.

*System Architecture.* The System Engineering effort will define the overall architecture of the REE System and will conduct a successful CDR of the flight prototype system design. It will establish a baseline architecture, and it will examine alternatives to this baseline. As part of this effort, it will examine alternative ways of meeting project milestones. The System Engineering effort will do this by flowing down project requirements and Project commitments to form a detailed set of system requirements, defining an implementation plan in coordination with the other task areas (Applications, Computing Testbeds, and System Software). It will incorporate a test engineering group to ensure that it meets the requirements as defined by system engineering. The System Engineering Team will provide for the development of system reliability and performability models and other system analysis tools as required.

### 6.4.2 System Integration, Testing and Validation

The System Engineering Team is responsible for the coordination and integration of the technical activities of the REE Project. The products of the Computing Testbeds and System Software activities must be coordinated so that requirements are appropriately distributed and addressed. System Engineering must also adjudicate requirements that Applications seek to impose on the system, and requirements the System Software and Computing Testbeds seek to impose upon Applications. Since most of the Project milestones involve products of all three of these activities, the System Engineering Team will serve as the integrator of these activities, testing and validating that the output metrics are met by the integrated product.

## 6.5 ADVANCED TECHNOLOGY OPPORTUNITIES

This effort targets the development of ultra-low-power processing and memory component prototypes. Several promising technologies exist which could have a high payoff in providing ultra-low-power systems. These technologies are all high-risk, because of their immaturity. The potential benefits, however, are quite large. Thus, REE will make a modest investment in an effort to develop an ultra-low-power computer using one of several high-risk/high-payoff technologies.

All of the low-power technologies under consideration owe their great promise to the same enabling technology trend: the ever-shrinking size of solid state device features (e.g., transistors, conducting paths). This trend is leading to lower operating voltages (and hence lower power) and to a greater density of gates that can be placed on silicon. As more gates can be placed on a single chip, so also can more functionality.

Ultimately, this trend will permit fully functional general purpose computers (including multiple CPUs, RAM, an interconnect structure, and off-chip drivers) to be placed on a single chip. This approach, sometimes referred to as Processor-In-Memory (PIM), has enormous advantages. First, the considerable power normally invested in moving data between chips (over 50% of the total in conventional architectures) is eliminated. Second, problems arising from memory bandwidth and latency, which invariably limit performance in conventional architectures, are dramatically reduced. Secondary benefits follow. For example, architectures may be simplified as the need for a complex cache structure is reduced. Possibly, caches will be eliminated altogether.

This trend will also permit the placement of high-functionality special-purpose computers on a single chip. The advantages of special purpose processors have been known for decades, with application-specific integrated circuits (ASICs) out-performing comparable-sized general-purpose processors by an order of magnitude or more. But ASICs come with an enormous disadvantage: they have no flexibility. Once fabricated and launched, an ASIC cannot be changed.

About a decade ago, field programmable gate arrays (FPGAs) were introduced, which added general-purpose flexibility to ASIC performance. But the low density of gates limited their functionality. As feature size continues to shrink, FPGA technology may be expanded, leading to a new generation of gate arrays with sufficient capacity for use in the general purpose arena. The customization available through FPGAs may ultimately prove to provide the best overall power efficiencies for given level of computing capability. But substantial investment is required in the development of tools to program FPGAs using high level languages, so that digital logic designers can be eliminated from the programming and testing loop.

In fiscal years 1997 and 1998, the REE Project made an initial investment in the development of PIM technology. Contracts were issued to Prof. Peter Kogge of Notre Dame to work with Lockheed Martin Federal Systems of Manassas, VA and insyte Corp. of Tampa, FL to develop and deliver a hardware PIM prototype to JPL. However, because of budget reductions in FY99, this effort was suspended.

Additional technology development opportunities may arise during the life of the Project. These may be software technologies as well as hardware technologies. The emerging System-On-a-Chip (SOC) technologies, for example, promise rapid and inexpensive development of custom architectures from off the shelf IP cores. There are additional development activities in the commercial sector aimed at providing software development tools for these SOCs. As these technologies develop, REE may study their use for fault tolerant parallel space-based computers. In addition, there is a considerable amount of applicable university research being done in these and related areas. The REE Project will periodically assess the return on its advanced technology investments and adjust its strategy as opportunities arise.

### 6.6 SUMMARY

As seen from the above technical summary, the REE Project will result in: a methodology for transitioning COTS components to space, a series of REE enabled science missions, and a first instantiation of an REE computer system ready for flight insertion. In addition, it will have resulted in two generations of REE computers (Scalable Testbed and Flight Prototype) and several years of experience in working with both commercial vendors and mission scientists. The Project thus will deliver a wealth of experience and proven capabilities by the time of its completion. It should also be noted that to maximize the impact of this technology on NASA’s future missions, a third and fourth generation should follow in rapid succession to keep up with COTS state-of-the-art computer systems. With each succeeding generation, system power performance will increase, while system development and fielding costs are reduced and newer, more powerful science missions enabled. Successful completion of the REE Project should spawn additional Agency efforts to maintain the state-of-the-art in NASA onboard computing.

## 7 Schedules

The REE Project has defined a series of Project milestones responsive to the updated HPCC Program milestones. These milestones are listed in Table 4. These milestones incorporate all of the unfulfilled milestones from the previous version of this plan (March 1999). They also include a number of additional milestones which further detail the progression of the Project towards its demonstration of spaceborne applications on a flight prototype embedded scalable system. Metrics to be applied for each milestone are defined in **Appendix B**. A table organizing these milestones by WBS may be found in **Appendix C**.

The REE Project Manager approves an Implementation Plan developed and maintained by the REE Chief Engineer in consultation with the WBS element Managers. This Implementation Plan contains the interim task milestones and integrated task schedules. WBS element Managers develop and maintain lower level schedules as needed. The Chief Engineer approves these schedules.

**Table 4. Milestones for the REE Project Organized by Program Milestone. Program Milestones are designated by HPCC x.x. Project Milestones are Numbered Chronologically by WBS Assignment.**

| Milestones | Due Date | Output Metrics |
|---|---|---|
| **HPCC 1.2 - Establish 1st generation scalable embedded computing testbed** | **6/01** | **Computing testbed capable of 30 MOPS/Watt and scalable to at least 50 nodes** |
| 2.2 - Testbed Upgrade Requirements Defined | 8/00 | Requirements documented in preparation for procurement |
| 4.2 - Preliminary System-wide Fault Model Defined | 12/00 | Preliminary orbit dependent fault model for major system components |
| 2.3 - 1st Generation Testbed Upgrade Installed | 3/01 | Enhancements satisfying testbed upgrade requirements installed |
| 2.4 - 1st Generation Testbed Complete | 6/01 | Benchmark Applications demonstrating 30 MOPS/W, architecture scalable to 50 nodes |
| **HPCC 2.1 - Develop real-time reliability for spaceborne computing** | **3/02** | **3 applications with 99% availability, 99% reliability over 5 years, and less than 50 msec latency.** |
| 4.1 - Initial Ground based Radiation Testing Completed | 9/00 | Total Integrated Dose (TID) and Single Event Upset (SEU) rates measured for PPC750 & secondary components |
| 3.1 - Non real-time Fault Tolerance Demonstration | 12/00 | 1 application with 99% availability, 99% reliability over 5 years |
| 3.2 - Preliminary SIFT Capability Demonstration | 3/01 | 3 applications with 99% availability and 99% reliability over 5 years |

| | | |
|---|---|---|
| 4.3 - Next Generation Processor Ground based Radiation Testing Completed | 9/01 | Total Integrated Dose (TID) and Single Event Upset (SEU) rates measured for next generation processor & secondary components |
| 3.3 - Real-time Fault Tolerance Demonstration | 12/01 | 1 real time application with 99% availability, 99% reliability over 5 years, 50 msec latency |
| 3.4 - Real-time SIFT Capability Demonstration | 3/02 | 3 applications (mixed real-time and non-real-time) with 99% availability, 99% reliability over 5 years, and less than 50 msec latency for real time applications. |
| **HPCC 5.1 – Demonstrate embedded applications on 1st generation spaceborne computing testbed** | **9/00** | **3 applications with 10X improvement (per processor) in throughput over the 1999 RAD6000, sqrt(n) processor scalability, and 50% of ideal speedup** |
| 1.1 - Initial Embedded Applications Demonstration | 3/00 | 3 apps, sqrt(n) scalability, 50% ideal speedup on Level Zero testbed |
| 2.1 - Install 1st Generation Testbed | 7/00 | 30 MOPS/W (benchmarks), scalable to 50 nodes |
| 1.2 - REE Science Applications operating on 1st Generation Testbed | 9/00 | 3 applications with 10X improvement (per processor) in throughput over the 1999 RAD6000, sqrt(n) processor scalability, and 50% of ideal speedup |
| **HPCC 5.4 - Demonstrate embedded applications using fault-tolerant techniques** | **6/02** | **3 applications with 10X improvement (per processor) in throughput over the 1999 RAD6000, sqrt(n) processor scalability, and 50% of ideal performance speedup** |
| 4.4 - Final System-wide Fault Model Defined | 12/01 | Final Orbit dependent fault model defined for major system components defined |
| 1.3 - Initial Embedded Application Demonstration using Fault Tolerance Techniques | 3/02 | 3 apps, 99% availability, 99% reliability over 5 years |
| 1.4 - Embedded Applications Demonstration using Fault Tolerance Techniques | 6/02 | 3 applications with 10X improvement (per processor) in throughput over the 1999 RAD6000 while operating with fault rates relevant to each application domain, sqrt(n) processor scalability, and 50% of ideal performance speedup |

| | | |
|---|---|---|
| **HPCC 6.2 – Establish impact on space mission through the demonstration of a flight-ready integrated system software, testbed, and application system** | **6/04** | **3 applications achieving 300 MOPS/Watt on flight qualified testbed with scalability to 50 nodes, scalability of sqrt(n), availability of 99%, reliability of 99% over 5 years, real time latency of less than 50 msec and price performance of 8 MOPS/$K (100X). Capability for insertion time of less than 18 months into flight vehicle. Programming environment which is the same as the equivalent commercial system.** |
| 4.5 - System Requirements Defined | 3/02 | Flight prototype hardware architecture and system software requirements document |
| 4.6 - Validation of System Design against Fault Model for Availability, Reliability | 9/02 | Successful CDR of flight prototype system design for 99% availability, 99% reliability over 5 years. |
| 3.5 - Integrated System Software 1st Delivery | 3/03 | SIFT layer prototype demonstrated on engineering model of flight prototype hardware |
| 3.6 - Integrated System Software Flight Delivery | 3/04 | SIFT layer integrated on flight prototype hardware |
| 2.5 - Flight Prototype Delivery | 3/04 | Flight qualified hardware delivered operating at 300 MOPS/W |
| 1.5 - Spaceborne Applications Demonstration | 6/04 | 3 applications achieving 300 MOPS/Watt on flight qualified testbed with scalability to 50 nodes, scalability of sqrt(n), availability of 99%, reliability of 99% over 5 years, real time latency of less than 50 msec and price performance of 8 MOPS/$K (100X). Capability for insertion time of less than 18 months into flight vehicle. Programming environment which is the same as the equivalent commercial system. |
| **HPCC 7.2 - Establish sustained utilization of commercial computing technologies for spaceborne applications** | **9/05** | **Technology selected for flight mission price performance of at least 8 MOPS/$K (100X).** |
| 6.1 - Flight Application Demonstration | 3/05 | Application selected for flight mission demonstrated on flight prototype or onboard |
| 6.2 - REE Technology Accepted for a Flight Mission | 9/05 | 1 mission baselines insertion of REE flight prototype or SIFT |

# 8 Resources

The resources section has been removed from this online version.

# 9 Controls

## 9.1 PROJECT PLAN CHANGES

The process for controlling changes to the REE Project and the subordinate WBS elements is hierarchical and described in this section.

The Program Commitment Agreement (PCA) is the overall controlling document for the HPCC Program, and REE as a constituent project. It is a contract between the NASA Administrator and the Associate Administrator of Aerospace Technology, defining the high level requirements and commitments for the HPCC Program and the REE Project. The HPCC Program Plan is the controlling document which defines the Program objectives and execution approach. The REE Project objectives and requirements are derived from this document. Any changes to the REE Project that affect the PCA or the HPCC Program Plan would require changes to the REE Project Plan and to all affected controlling documents. The process for making and approving changes to these documents is detailed in the HPCC Program Plan.

Changes within the REE Project which impact the Project objectives, technical scope, schedule, or budget guidelines but do not impact higher level controlling documents require the approval of the HPCC Program Manager and JPL Center Director, and are captured in a revised REE Project Plan.

For changes to the REE Project within the objectives, technical scope, schedule and budget guidelines established in the approved Project Plan, the REE Project Manager has the authority to approve such changes. Such changes are captured in a revised Project Implementation Plan.

A formal process is used for managing Project changes: requesting, acquiring the required level of approval, and tracking and documenting the changes. The REE Project Manager maintains the Project change log.

## 9.2 COMPUTING TESTBEDS

All participants of the REE Project must comply with the NASA policy on access to software, data, and testbed facilities. Access to the REE testbeds will be open to U.S. citizens and U.S. permanent resident aliens. Access to the REE testbeds by foreign nationals requires advanced approval regardless of whether the foreign national is approved for physical access to JPL. The JPL Legislative and International Affairs Office is responsible for the initial foreign national approval process.

## 9.3 SENSITIVE TECHNOLOGY

The Government, and the California Institute of Technology (Caltech), shall have unlimited rights to technical data and computer software produced in the performance of contracts issued by JPL under the NASA REE Project. (Caltech operates JPL under contract from NASA.) Unlimited rights, as used above, means the right to use, disclose, reproduce, prepare derivative works, distribute copies to the public, and perform publicly and display publicly in any manner and for any purpose, and to have or permit others to do so. These unlimited rights extend to the use of technical data contained in proposals upon which such contracts are based. Technical data and computer software developed at private expense, including minor modifications thereof, remain the property of the developing entity and are protected from unauthorized disclosure and use. Government rights and the rights of Caltech are defined by the JPL Prime Contract with NASA, which governs all activities undertaken by JPL.

All information released by JPL outside of JPL will be done in accordance with JPL Policy: *Releasing Information Outside of JPL.* The release to a foreign national of technical information that resides at or is controlled by JPL requires advanced approval through the JPL Legislative and International Affairs Office as described in Section 9.2 above. The REE Project implements security techniques which prevent access to critical technology from "open" exchange systems and networks, and complies with JPL Information Technology security policies and requirements.

Negotiated License Agreements are used to restrict access to privately developed technology performed under the auspices of the REE Project. These agreements provide NASA with limited rights to use proprietary data or designs in NASA in-house or cooperative research projects. These agreements specify limits on the distribution and use of the proprietary data by NASA and NASA-licensed entities.

Some sensitive information developed solely within the REE Project may be subject to protection under the Export Administration Regulations or the International Traffic in Arms Regulations, which are export controls established by law. The participants in the REE Project will follow applicable export control laws. These regulations establish lists or categories of technical data and/or products that may not be exported without an approved export license. (Note that the definition of "exported" includes "disclosed" and "discussed" as well as published.)

Technical data and computer software produced for REE by another NASA Center is governed by each Center’s Policies and Procedures for the control of Sensitive Technology. Work performed at other NASA Centers shall comply with that Center’s Policies and Procedures, and applicable Federal Law.

# 10 Implementation Approach

The development of the Project hardware deliverables will be done largely out-of-house. RFPs were issued for the execution of a Study Phase and for the fabrication and delivery of a First Generation Testbed. An RFP will be issued for the eventual development of a flight prototype. The development of Project system software deliverables will be done in partnership with industry, academia, and other government agencies. External contracts will be submitted to competitive bidding to the maximum extent practical. The development of algorithms and software for applications will be led by NASA scientists or mission managers, with the work performed largely at their home institutions.

## 10.1 REE WBS

A Work Breakdown Structure (WBS) has been developed to reflect the major activities being undertaken over the life of the Project. This WBS is organized around the principle technical activities of the Project, as detailed in the **Technical Summary** section, and the cross-cutting functions of Project Management and System Engineering. The activities under Advanced Technology Investigations are fluid and additional activities may be added during the life of the Project, at the discretion of the Project Manager.

1.0 Spaceborne Applications

- 1.1 Science Applications
- 1.2 Applications Technical Support
- 1.3 Application-Based Fault Tolerance

2.0 Embedded Computing Hardware Research and Development

- 2.1 Testbeds
- 2.2 Early Prototype

3.0 System Software Research and Development

- 3.1 Software Implemented Fault Tolerance Architecture
- 3.2 Software Implemented Fault Tolerance Development
- 3.3 Prototype System Software

4.0 System Engineering

- 4.1 Studies
- 4.2 Modeling
- 4.3 System Design
- 4.4 Validation and Test

5.0 Advanced Technology Investigations

- 5.1 Processor In Memory (PIM)
- 5.2 Node Controller ASIC

6.0 Management

- 6.1 Project Management
- 6.2 Education and Outreach

## 10.2 PROJECT DESCOPE PROCESS

Should descoping of the REE Project or rescoping of any of its constituent WBS elements be required, whether due to resource reductions in the REE Project or the need to rebalance the resources within the Project, the following descope process will be followed.

1. The REE Project Manager, in consultation with the Chief Engineer and element Managers, will develop a list of current Project activities on the critical path for Project Milestones. Each element Manager will define the minimum level of activity required to adhere to schedule.

2. The Project Manager will rebalance the available resources to maintain schedule at the expense of increased risk of failure to achieve Project milestones on time. Risk to testbed milestones will be increased first, application milestones second, and system software milestones last.
3. If schedule cannot be maintained with the available resources, the Project Manager will attempt to reschedule Project Milestones to conform to the expected resources profile, and request approval from the HPCC Program Manager.
4. If rescheduling Project Milestones is not possible under the expected resources profile, the Project Manager will propose a new set of Project Milestones which correspond to reduction in demonstrated system capability at the end of the Project, and request approval from the HPCC Program Manager. Hardware performance will be targeted first, followed by real time SIFT capability.
5. In the event that the available resources no longer support the development of SIFT capable of handling the fault rates in low Earth orbit or deep space, the Project Manager shall recommend to Program Management that the REE Project be canceled.

## 11 Acquisition Summary

Free and open competitive procurements will be used to the maximum extent possible. The primary procurement vehicle expected to be used in the REE Project is the Request for Proposals (RFP). At JPL, this vehicle results in contracts. Interagency agreements for joint R&D endeavors may also be used as the occasion arises.

## 12 Project Dependencies

The desired spaceborne demonstration of an REE system is dependent on the identification of a flight opportunity in which the launch and operations costs are borne by some other project. These costs are not budgeted within REE. Achievement of the final REE Project Milestones does not require a flight opportunity. However, the acceptance and infusion of REE technology into other projects will be more likely if an opportunity can be exploited.

## 13 Agreements

There are no signed Project agreements as of this writing. A Memorandum Of Understanding (MOU) between REE and the AFRL’s ISCP is currently being negotiated. This MOU will cover joint development of software, sensor interfaces, and secondary storage capabilities on the REE First Generation Testbed and ISCP architecture. ISCP and REE have independently competitively selected the same contractor for the current phase of each project. The MOU seeks to prevent duplication of work and expansion of the technical development made possible by a common prime contractor.

# 14 Performance Assurance

The REE Chief Engineer is responsible for performance assurance of all deliverables. The Chief Engineer will employ standard JPL performance assurance processes to test and validate all software and hardware deliverables.

# 15 Risk Management

The HPCC Program maintains a Risk Management Plan at the Program Level. The REE Project adheres to the requirements of that plan, and subscribes to the risk management process detailed in it. This section delineates the risks which are specific to the REE Project, and the Project's processes for mitigating those risks.

Risk can be classified into two general categories: *technical* risk and *resource/schedule* risk. The first refers to uncertainty arising from unexpected development difficulties. The REE Project has been structured to minimize the risk associated with the attainment of Project milestones and their minimum success criteria. While we expect to meet these criteria, there are in addition several "stretch goals," high-payoff/high-risk elements for which success will substantially exceed Project commitments. The second risk category, *resource/schedule* risk, involves factors that are programmatic in nature.

## 15.1 TECHNICAL RISK

There are three primary technical risks facing the REE Project:

1) That reductions in power for device component technology will not attain the expected industry projections for the year 2004.
2) That software-implemented fault tolerance will not prove sufficiently reliable to permit the extensive use of COTS-based technologies.
3) That hardware or software subcontractors fail to meet delivery schedules or performance specifications.

The impact of (1) could be the failure of the Project to meet the performance criteria for Project Milestones in 2004. The consequence of (2) is that REE would be forced to include at least some radiation-hardened components in the flight prototype, again lowering performance. In addition, cost would be increased. Occurrence of (3) would likely cause delays in the achievement of Project Milestones in 2004.

The REE Project will mitigate the first risk by making strategic investments in alternative ultra-low power technologies. Several promising, but immature, technologies have the potential for revolutionary breakthroughs in power performance. The key enabling technology for all of these is the dramatic increase in the density of gates that can be implemented on silicon. This trend may permit the placement of fully functional general purpose computers or reconfigurable special purpose computers on a single chip. The elimination of the power normally required to move data off-chip and between chips would represent a significant improvement and could provide REE with an alternative path to the targeted power performance.

The REE Project will mitigate the second risk by leveraging related programs managed by the Air Force and by DARPA. The Air Force Improved Space Computer Program (ISCP) has placed a high premium on system survivability. Consequently, ISCP will invest a major portion of its resources in the development of radiation-hardened components. REE will coordinate its own milestones and investment strategy with ISCP to leverage this development and provide REE with an alternative path for radiation-tolerance. If necessary, REE will incorporate radiation hardened components in critical sections of the architecture to raise the overall system reliability to the required level. A second mitigation strategy is to incorporate replicated, voted components into the architecture to achieve the required system reliability.

Mitigation of the third risk is based on prudent subcontracting practice, and close interaction with subcontractors during the contract execution. Early knowledge and awareness of technical difficulties in performance of contracts will allow the Project to assist the contractor in taking corrective action, and to implement alternative strategies, including going to a second source, to prevent serious cost and schedule impacts. Sufficient schedule margin has been built into the Project to allow for some delivery delays.

**Table 7. Technical Risk Assessment.**

| **Risk** | **Risk Impact** | **Risk Probability** | **Risk Probability Mitigation Process** |
|---|---|---|---|
| Component technologies do not attain power and performance capabilities projected by industry for 2004 | High | Low | · Invest in alternative ultra-low-power technologies |
| SIFT technology does not attain sufficient reliability to permit the extensive use of COTS in space | High | Medium | · Allow for replicated/voted components in critical sections of the architecture of the flight prototype<br>· Leverage related programs managed by the Air Force and DARPA to incorporate radiation-hardened components into critical sections of the architecture |

| | | | |
|---|---|---|---|
| Subcontractors fail to meet delivery schedules or performance specifications | Medium | Medium | · Closely monitor contractor performance and development critical path progress<br>· Plan sufficient schedule margin to absorb minor delivery delays<br>· Maintain a viable second source option which can be exercised quickly if a serious performance delay is encountered |

## 15.2 PROGRAMMATIC RISK

There are three primary programmatic risks facing the REE Project:

1) That the end result of the REE Project will not be adopted by future NASA missions.
2) That the private sector developers of state-of-the-art software will not allow the REE prime contractor(s) to license and modify their software.
3) That the REE Project could suffer a reduction in the resources available to meet the Project’s commitments.

A major concern to the REE Project is that many technology development projects result in technology advances which are not successfully transferred to the intended beneficiaries. This occurs for a variety of reasons, with the primary reason being a lack of attention to the customers needs during the project development. The REE Project is structured to mitigate this risk by engaging the intended customer base (mission science Principle Investigators and mission project managers) from the very beginning of the project. Through the REE Applications Teams, the Project will continuously feed the science missions’ needs and requirements into the hardware and software development efforts, so that the end software and hardware technology developed during the Project is driven by and is consistent with the customers’ needs for enhanced mission science return at reduced cost. In addition, REE will collaborate and coordinate as appropriate with advanced avionics and flight software development activities to ensure interoperability and compatibility so that insertion into flight systems will be seamless and straightforward.

The consequence of the second risk would be the exclusion of the spaceborne community from the use of popular commercial products, including programming environments, tools, and debugging software. The REE Project will mitigate this risk by minimizing the need to modify COTS software to support software implemented fault tolerance, maintain active relationships with leading COTS operating systems developers, and examine the use of open-source tools and operating systems.

Resource reduction is an area of relatively high risk to the REE Project. Annually (and sometimes more often) the Project faces challenges to its budget from all levels of management and oversight. The REE Project has outlined descope options that can accommodate modest resource reductions, while maintaining the overall goals of the Project. For example, the capturing of a flight opportunity for engineering/science demonstration of REE technology is a "stretch goal." A modest reduction in resources could put this goal at risk. However, the elimination of a flight demonstration does not pose a risk to any Project Milestones, since REE does not require a flight in order to satisfy its commitments to the Program. In the case of severe reductions, changes to Project Milestones will be proposed in a revised project plan.

**Table 8. Programmatic Risk Assessment.**

| Risk | Risk Impact | Risk Probability | Risk Probability Mitigation Process |
|---|---|---|---|
| REE technology transfer unsuccessful | High | Medium | · Involve principal REE customer base (instrument scientists) from inception of the project<br>· Continuously feed science-driven requirements into the hardware and software development efforts.<br>· Ensure interoperability and compatibility with next generation avionics hardware/software systems |
| Private sector developers of software will not allow prime contractor(s) to license or modify their software | Medium | Medium | · Design SIFT layers to minimize need to modify COTS software<br>· Maintain active relationships with leading COTS operating system developers |
| Reduction in funding | High | Medium | · Advocate benefits to customers/stakeholders<br>· Maintain agile project descope plan |

## 16 Environmental Impact

The Environmental Impact procedures and guidelines are not applicable to the REE Project.

## 17 Safety

The Safety procedures and guidelines are not applicable to the REE Project.

# 18 Technology Assessment

The REE Project is a computer research project that pursues technologies that are between five and ten years from maturity. Applications in the areas of Earth and space science are used as drivers of REE's technology research, providing the requirements context for the work that is done.

REE conducts TRL 2–6 research activities intended to prove feasibility, develop and demonstrate computing technologies for eventual introduction into NASA operations though entities such as New Millennium, Discovery, Shuttle and Space Station. REE work in spaceborne COTS parallel computing systems is now at the TRL 2-3 stage, but is planned to attain TRL 6 in 2004.

# 19 Commercialization

JPL is committed to transferring its technology to the private sector. The following vehicles are available for commercialization of technology, and the REE Project will utilize them depending on mission need and resources.

> **Technology Affiliates:** JPL transfers technology and expertise to U.S. companies on a reimbursable basis to solve key problems identified by the company.
>
> **Strategic Technology Development Alliances:** JPL develops commercial R&D alliances with U.S. industry focused on shared investment, risk, and benefit strategies.
>
> **Targeted Commercialization:** JPL targets the commercialization of its validated technologies into emerging global markets.
>
> **New Venture Spin-Offs:** JPL enables spin-off/start-up companies from the JPL technology base.
>
> **Participation in Federal/State Technology Initiatives:** JPL establishes a strategic presence in National/State technology initiatives where JPL's technology base will be leveraged for U.S. economic competitiveness and related policy goals.
>
> **Regional Economic Growth**: JPL encourages economic growth in the region.

In addition, the REE Project will sponsor and conduct technical meetings and workshops and promote the publication of scientific and technical papers to maintain the flow of technology from NASA to industry and academia.

# 20 Reviews

In fiscal year 2000, the REE Project will form a technical review board. This board is chartered to review the progress and plans of the Project for consistency, feasibility, and compatibility with spacecraft architecture constraints. The Board will meet at least once annual to advise the Project, and may meet more frequently as the need arises.

The REE Project is subject to Independent Annual Reviews (IARs). These are conducted as part of an overall IAR of the HPCC Program and of the other Projects in the program.

Technical reviews of each Project convened by the HPCC Program are conducted annually. Typically, these consist of end-of-year site reviews at the Project Lead Centers.

The REE Project Manager reports performance monthly to the HPCC Program Office and to the Office of Space Science (Code S).

The REE Project routinely generates the following reports:

REE Project Annual Report

Project Monthly Reports

Each of REE's element Managers report status and accomplishments on a monthly basis to the Deputy Project Manager, who synthesizes these reports into the Project Monthly Report.

## 21 Tailoring

The REE Project will be managed and implemented in accordance with the normal procedures used by the Jet Propulsion Laboratory for technology development activities, and in compliance with all requirements established by law and regulations. Executive orders and Agency directives will be observed to the extent accepted by the JPL Prime Contract. There are no major deviations from these procedures.

This Project Plan has been tailored to address the specific needs of an advanced technology research activity. This activity has overarching goals and objectives, but due to the natural uncertainty of any research activity, the specific systems, technical specifications, and end product description and operation are not fully developed. Certain sections of this plan (Technical Summary, Schedules, Implementation Approach, Acquisition Summary, Risk Management, and Technology Assessment) have been tailored in their content. Subsections that are more appropriate to a flight project or other major systems project have been eliminated as not relevant to the research nature of this project. Other sections (Environmental Impact and Safety) have no special significance to this project.

## 22 Change Log

May 1997 1st Approved REE Project Plan

Mar 1999 Project Plan revisions to accommodate funding reduction in FY99 and FY00. Plan structure and content revised to conform to NPG 7120.5A and new WBS structure developed to more closely align with the Project major activities

1. Computing Testbeds milestone CT 8 is delayed from 03/99 to 12/99 due to funding reduction in FY99. Low power technology studies suspended for FY99.
2. Grand Challenge Applications milestone GC 6 is delayed from 06/99 to 03/00. The completion of this milestone depends on the completion of CT 8
3. System Software milestone SS 5 is delayed from 03/00 to 09/00. The completion of this milestone depends on the completion of CT 8.

Mar 2000 Project schedule revised to add an experimentation phase between the delivery of the FGT and the fabrication of the flight prototype. Schedule revisions to accommodate changes in HPCC Program PCA and Program Milestones. Project Milestones renumbered to conform to WBS designations. Resource tables updated to reflect additional planned work at GSFC and Program Office support at ARC.

1. Milestone CT 8 is renumbered 2.1 and is delayed from 12/99 to 7/00 due to contractor fabrication problems with the FGT.
2. Milestone GC 6 is split into two milestones: 1.1 due 3/00 as previously scheduled, but now to be achieved on a locally implemented testbed due to the slip in the FGT, and 1.2 due 9/00 to complete the demonstration using the contractor delivered testbed.
3. Milestone SS 5 is renumbered 3.2 and is delayed to 3/01 due to the slip in the FGT.
4. Milestone SS 6 is renumbered 3.4 and is delayed to 3/02.
5. Milestone CT 10 is renumbered 2.5 and is rescheduled for 3/04 to accommodate the addition of the experimentation phase.
6. Milestone GC 8 is renumbered 1.5 and is rescheduled for 6/04 to accommodate the addition of the experimentation phase.
7. Additional Project Milestones are defined to reflect the increased emphasis on system engineering and to address the new set of HPCC Program Milestones for which REE is responsible.
8. A Project Scientist role is added and the responsibilities of other key Project Staff are clarified

# Appendix A
# Acronyms

| | |
|---|---|
| ABFT | Algorithm-Based Fault Tolerance |
| AFRL | Air Force Research Laboratory |
| API | Application Program Interface |
| ARC | Ames Research Center |
| ASIC | Application Specific Integrated Circuit |
| Caltech | California Institute of Technology |
| CAS | Computational Aerospace Sciences |
| CDR | Critical Design Review |
| CIC | Computing Information and Communications |
| COTS | Commercial Off-the-Shelf |
| CPU | Central Processing Unit |
| DARPA | Defense Advanced Research Projects Agency |
| DOD | Department of Defense |
| ESS | Earth and Space Sciences |
| ESSP | Earth System Science Pathfinder |
| FGT | First Generation Testbed |
| FPGA | Field Programmable Gate Array |
| FTE | Full Time Equivalent |
| FY | Fiscal Year |
| GB | Giga ($10^9$) Byte (of memory) |
| GeV | Giga ($10^9$) Electron Volts |
| GFLOPS | Giga ($10^9$) Floating Point Operations Per Second |
| GLAST | Gamma-Ray Large Area Space Telescope |
| GOPS | Giga ($10^9$) Operations Per Second |
| GSFC | Goddard Space Flight Center |
| HP | Hewlitt Packard |
| HPC | High Performance Computing |
| HPCC | High Performance Computing and Communications |
| IAR | Independent Annual Review |
| I/O | Input/Output |
| IP | Intellectual Property |
| ISCP | Improved Space Computer Program |
| JPL | Jet Propulsion Laboratory |
| KOPS | Kilo (Thousand) Operations per Second |
| LT | Learning Technologies |
| MeV | Million ($10^6$) Electron Volts |
| MIPS | Million Instructions Per Second |
| msec | milliseconds |
| µm | micro ($10^{-6}$) meter (micron) |
| MOPS | Millions of Operations Per Second |

| | |
|---|---|
| MOU | Memorandum of Understanding |
| MPI | Message Passing Interface |
| NASA | National Aeronautics and Space Administration |
| NGST | Next Generation Space Telescope |
| NPG | NASA Procedures and Guidelines |
| NREN | NASA Research and Education Network |
| OS | Operating System |
| OTIS | Orbiting Thermal Imaging Spectrometer |
| PCA | Program Commitment Agreement |
| PIM | Processor-in-Memory |
| R&D | Research and Development |
| RAM | Random Access Memory |
| REE | Remote Exploration and Experimentation |
| RFP | Request for Proposal |
| SEU | Single Event Upset |
| SGI | Silicon Graphics, Inc. |
| SIFT | Software-Implemented Fault tolerance |
| SOC | System-On-a-Chip |
| TeV | Tera ($10^{12}$) Electron Volts |
| TID | Total Integrated Dose |
| TRL | Technology Readiness Level |
| VNIR | Visible/Near Infrared |
| WBS | Work Breakdown Structure |

# Appendix B
# REE Project Metrics

This section details metrics that have been established for measuring practical progress toward the REE Project Objectives. These metrics have been developed cooperatively between the Program and Project offices. They will be actively used for evaluation, management, and reporting.

The Project milestones have been constructed to demonstrate steady progress towards the achievement of a practical scalable embedded computing environment for NASA applications. These milestones can be categorized into distinct aspects of the conditions necessary for the Project to be deemed successful: embedded application performance, hardware power performance and usability, system software portability, and overall system reliability. Taken together, achievement of these milestones will constitute de facto achievement of the practical embedded scalable computing environment for space that is the Project's goal. Metrics detailed here will be used to determine when a milestone has been successfully completed and to monitor progress towards achieving each milestone.

The Project milestones express achievements in two broad categories: performance and usability. Each requires different measurement tools and different environment considerations. The performance aspects of milestones are generally straightforward to measure. Usability is more difficult to measure, since the characteristics of usability often are specific to the functionality of a particular piece of software or hardware. Certain general characteristics for high performance systems are necessary conditions of usability and are quantifiable and measurable. These are scalability and speedup. Together with portability, these constitute the primary metrics for the applications and system software.

| | | |
|---|---|---|
| 1 | Scalability | reflects the need to execute as large an application configuration as possible in the same elapsed time on different sizes of parallel computing platform configurations with no additional development effort |
| 2 | Speedup | reflects the need to execute a specific application configuration in the least amount of time by using multiple processors |
| 3 | Throughput | reflects the ability to execute an application in the least amount of time. |
| 4 | Portability | reflects the practical need for software to be developed on a commercially available system and executed on an embedded system which may not be available until late in the software development cycle or to outlive the effective lifetime of the current generation of HPC systems |
| 5 | Power-Performance | reflects the stringent limits on power for spaceborne and highly portable or remote earth-based systems. It also reflects the fact that performance requirements may actually be increased from those of non-miniaturized systems. |

| | | |
|---|---|---|
| 6 | Reliability | reflects the need to have a very high probability of sustained correct operation over long periods of time, including unattended operation in the presence of faults. |
| 7 | Availability | reflects the need to have a very high probability that the system will function properly at a given moment, including unattended operation in the presence of faults. |

These seven metrics are defined and the rules for their use are described in the paragraphs below. Their order in the above table does not represent their relative importance, nor will all metrics be applied to every milestone. It is important to note that these metrics have meaning only in terms of a *platform and application* taken together. Hence, these metrics will not be used to rank or evaluate "bare" platforms independent of application software. Instead, evaluations of system configurations will be made using benchmark kernels that are representative of actual REE Project algorithms. Indeed, the primary use of these metrics will not be to evaluate computers at all, but to define success criteria for specific Project milestones, where associated platforms and applications are clearly defined.

### ❑ Scalability

Applications and platforms need to be able to execute efficiently in a variety of configuration sizes without re-engineering. This characteristic is referred to as ***scalability***. This metric is derived from the practical requirement that development costs effectively prohibit either software or hardware from being problem size specific. The economies of high performance computing demand that both software and hardware need to be able to function without change on small, medium, and large problems. Scalability has slightly different meanings when applied to software or hardware.

Software scalability refers to the ability of an application or tool to execute work proportional to platform size with a bounded growth in execution time. This concept is best illustrated by an example. Consider the application of counting the number of zeros in a dataset. Suppose the application takes 10 seconds to accomplish the task for a given dataset size on a single processor. If the application also takes 10 seconds to count the zeros in a dataset 50 times as large on a parallel computer with 50 processors, the software is said to be perfectly scalable. It has no growth in execution time as the problem size is increased proportional to the machine configuration size. As a practical matter, some execution time growth is tolerable if substantially larger applications are enabled. Therefore, the Project will consider software to be scalable if execution time growth for scaled applications is no worse than square root (*sqrt*) of machine configuration size. Scalability is a dimensionless parameter defined as:

$$S_c(n) = T_{n\times w}/T_w$$

where $T_{n\times w}$ is the execution time for an application which does $n\times w$ work on $n$ processors and $T_w$ is the execution time doing $w$ work on a single processor. Thus, the scalability metric is satisfied if

$$S_c(n) \leq \mathrm{sqrt}(n)$$

is achieved over a sufficient range of n.

Hardware scalability refers to the ability to assemble functioning platform configurations with the same programming and execution environment and reasonable Mean Time Between Failures in a variety of sizes. The largest configuration size for which these conditions hold is deemed to be the scalability limit. The Project will defer to manufacturers designations of their largest product configuration, absent evidence to the contrary.

### ❑ Speedup

For certain problem classes, absolute time to solution is more important than scalability. ***Speedup*** measures the proportional decrease in execution time for a fixed problem as a function of machine configuration size. Speedup is a dimensionless parameter defined as a ratio of execution times:

$$S_p(n) = t_1/t_n$$

where $t_1$ is the problem execution time on a single processor and $t_n$ is the execution time on *n* processors. Because a fixed problem, by definition, has a predetermined limit to the number of operations it performs, its speedup will always have an upper bound.

### ❑ Throughput

To compare a given application on multiple platforms, ***throughput*** can be used as a metric of the number of times the application can be executed in a given time period. It is usually measured as the inverse of execution time ratio for a given application run on two different machines, and can be defined as:

$$T_{2,1}= t_1/t_2$$

where $t_1$ is the amount of time needed to run the application on machine 1, and $t_2$ is the amount of time needed to run the application on machine 2. $T_{2,1}$ is then the throughput increase for machine 2 against a baseline of machine 1.

### ❑ Portability

To preserve the value of the initial development investment in an application, ***portability*** of software among the major vendors' platforms is an important attribute of the software design and the execution environment. Portability in the strict sense simply means being able to move an application from one platform to another and have it execute correctly with only a recompile and relink. This implies that the source language(s) is(are) available and that the runtime environment (libraries, OS interfaces, files system interfaces) is the same across platforms. An additional consideration is that the ported application exhibit similar efficiencies (scalability, speedup, performance) on the new platform as on the old. For the purpose of this document, portability is defined as a logical parameter which assume the values “true” and “false.” Software which does not require detailed knowledge of the operating system behavior and of hardware configuration will be considered “portable” if it requires no more than name replacements and argument list changes to make it run on a new platform.

For software that requires detailed knowledge of operating system behavior and of hardware configuration, the definition of portability must be relaxed to allow for the construction of custom drivers and interfaces to match the hardware and OS functionality. The software implemented fault tolerance layers will fall into this category. For this class of software, portability will be defined as requiring no more than the replacement of drivers and interfaces totaling less than 10% of the total number of lines of code.

### ❑ Power Performance

There are stringent limits on the power, mass, and size of systems that are launched into space or developed for highly portable earth-based applications (e.g., laptop computers). At the same time performance requirements may actually be *increased* over those of past missions (or earlier generation laptops). The power performance metrics characterize the ability of a flight system to attain a given performance level per unit electrical power. We will not specifically address the issues of mass, and volume, but expect that commensurate improvements will naturally result from improvement in power performance. In actual experience it is most often the limitation on power that limits performance. Power performance is measured in MOPS/watt, where MOPS is Millions of Operations Per Second (which may be a mixture of 32 bit integer and floating point arithmetic or logical operations). Although MIPS (Millions of Instructions per Second) is a more traditional measure of processor capability, it does not quantify the actual amount of work accomplished on processors which have complex instruction sets. In many cases, though, MOPS and MIPS may be interchangeable.

### ❑ Reliability

Reliability is defined as the probability of “correct operation” up to time t = T given that the system was operating correctly at time t = 0. “Correct operation” is defined as the absence of any fault condition from which the system cannot recover. Partial loss of capability following fault recovery may or may not constitute the loss of correct operation. Reliability can assume values from 0 to 1. Flight subsystem design specifications invariably call for reliabilities very close to 1.

System reliability is exceedingly important for spaceborne applications for the simple reason that a flight computer, once launched, cannot be repaired or replaced. Reliability characterizes the ability of a flight computer to recover from fault conditions (or avoid them altogether), which arise mostly from high levels of radiation. Fault recovery in reliable systems will be achieved with limited loss of performance. Reliability is an overarching metric which encompasses several other familiar attributes of flight computing systems, including fault tolerance and graceful degradation.

### ❑ Availability

Availability $A_v(t)$ is defined as the probability of correct operation at time t = T. Availability differs from reliability in that it contains no requirement regarding correct operation in the past. That is, it is an instantaneous (or differential) probability associated with an instant in time, rather than an aggregate (or integral) probability associated with an extended period of time. Availability may be affected both by the occurrence of fault conditions and by competition for system resources by multiple users. Availability can assume values from 0 to 1.

System high availability is particularly important for spacecraft operations tasks requiring real-time response where values of $A_v(t)=1$ may be required for mission success or safety. Conversely, for spaceborne science instruments characterized by high output bandwidth, availability (and reliability) may be traded off against speedup to maximize science return. The

REE architecture will enable spacecraft engineers and instrument scientists to allocate system resources to make these trades, based on an assessment of their particular requirements.

### ❑ Other Computing Milestone Metrics

Some milestones require additional metrics which are specific to the milestone.  In most cases, these are success counts.  They may be specific numbers or a percentage of maximum possible, depending on the milestone, and are indicative of success across a variety of types of applications.

Real time latency is defined as the ability to respond to an external event, such as an externally generated signal to the system, and take appropriate action within a specified period of time. Because the amount of time required to execute a signal handler depends on the details of the handler itself, real time latency is defined here to be the elapsed time between the time an externally generated signal is input to the systems and the time at which the first instruction of the signal handler is executed.  This latency is to be accomplished in the presence of faults at rates expected in low Earth orbit.

Each Project milestone will generally require two or more metrics against which progress will be measured.  This is due to the complex nature of each of the milestones, and the fact that most milestones require the demonstration of both usability and performance.  A milestone will be considered completed when the success criteria for all of the metrics applied have been met or exceeded.

# Appendix C
# Project Milestones Organized By WBS

| Milestones | Due Date | Output Metrics |
|---|---|---|
| **1. Spaceborne Applications** | | |
| 1.1 - Initial Embedded Applications Demonstration | 3/00 | 3 applications, sqrt(n) scalability, 50% ideal speedup on Level Zero testbed |
| 1.2 - REE Science Applications operating on 1st Generation Testbed | 9/00 | 3 applications with 10X improvement (per processor) in throughput over the 1999 RAD6000, sqrt(n) processor scalability, and 50% of ideal speedup |
| 1.3 - Initial Embedded Application Demonstration using Fault Tolerance Techniques | 3/02 | 3 applications, 99% availability, 99% reliability over 5 years |
| 1.4 - Embedded Applications Demonstration using Fault Tolerance Techniques | 6/02 | 3 applications with 10X improvement (per processor) in throughput over the 1999 RAD6000 while operating with fault rates relevant to each application domain, sqrt(n) processor scalability, and 50% of ideal performance speedup |
| 1.5 - Spaceborne Applications Demonstration | 6/04 | 3 applications achieving 300 MOPS/Watt on flight qualified testbed with scalability to 50 nodes, scalability of sqrt(n), availability of 99%, reliability of 99% over 5 years, real time latency of less than 50 msec and price performance of 8 MOPS/$K (100X). Capability for insertion time of less than 18 months into flight vehicle |
| **2. Embedded Computing Hardware Research and Development** | | |
| 2.1 - Install 1st Generation Testbed | 7/00 | 30 MOPS/W (benchmarks), scalable to 50 nodes |
| 2.2 - Testbed Upgrade Requirements Defined | 8/00 | Requirements documented in preparation for procurement |
| 2.3 - 1st Generation Testbed | 3/01 | Enhancements satisfying testbed upgrade |

| | | |
|---|---|---|
| Upgrade Installed | | requirements installed |
| 2.4 - 1st Generation Testbed Complete | 6/01 | Benchmark Applications demonstrating 30 MOPS/W, architecture scalable to 50 nodes |
| 2.5 - Flight Prototype Delivery | 3/04 | Flight qualified hardware delivered operating at 300 MOPS/W |
| **3. System Software Research and Development** | | |
| 3.1 - Non real-time Fault Tolerance Demonstration | 12/00 | 1 application with 99% availability, 99% reliability over 5 years |
| 3.2 - Preliminary SIFT Capability Demonstration | 3/01 | 3 applications with 99% availability and 99% reliability over 5 years |
| 3.3 - Real-time Fault Tolerance Demonstration | 12/01 | 1 real time application with 99% availability, 99% reliability over 5 years, 50 msec latency |
| 3.4 - Real-time SIFT Capability Demonstration | 3/02 | 3 applications (mixed real-time and non-real-time) with 99% availability, 99% reliability over 5 years, and less than 50 msec latency. |
| 3.5 - Integrated System Software 1st Delivery | 3/03 | SIFT layer prototype demonstrated on engineering model of flight prototype hardware |
| 3.6 - Integrated System Software Flight Delivery | 3/04 | SIFT layer integrated on flight prototype hardware |
| **4. System Engineering** | | |
| 4.1 - Initial Ground based Radiation Testing Completed | 9/00 | Total Integrated Dose (TID) and Single Event Upset (SEU) rates measured for PPC750 & secondary components |
| 4.2 - Preliminary System-wide Fault Model Defined | 12/00 | Preliminary orbit dependent fault model for major system components |
| 4.3 - Next Generation Processor Ground based Radiation Testing Completed | 9/01 | Total Integrated Dose (TID) and Single Event Upset (SEU) rates measured for next generation processor & secondary components |
| 4.4 - Final System-wide Fault Model Defined | 12/01 | Final Orbit dependent fault model defined for major system components defined |
| 4.5 - System Requirements Defined | 3/02 | Flight prototype hardware architecture and system software requirements document |

| | | |
|---|---|---|
| 4.6 - Validation of System Design against Fault Model for Availability, Reliability | 9/02 | Successful CDR of flight prototype system design for 99% availability, 99% reliability over 5 years. |
| **6. Project Legacy** | | |
| 6.1 - Flight Application Demonstration | 3/05 | Application selected for flight mission demonstrated on flight prototype or onboard |
| 6.2 - REE Technology Accepted for a Flight Mission | 9/05 | 1 mission baselines insertion of REE flight prototype or SIFT |

# Appendix D
# Description of the Five Current REE Applications

## Gamma-ray Large Area Space Telescope (GLAST)

*Principal Investigator: Prof. Thompson Burnett (University of Washington)*

The Gamma-ray Large Area Space Telescope (GLAST) is a next-generation high-energy gamma-ray telescope that will operate in the energy range from 10 MeV to 300 GeV. GLAST is currently part of NASA's Office of Space Science Structure and Evolution of the Universe program strategic plan. The GLAST mission is based on a new pair-conversion telescope design that utilizes modern solid-state particle detector tracking technology (i.e., silicon-strip detectors). To realize the *full* scientific potential of the GLAST instrument will require substantial on-orbit supercomputing resources (about 5 GOPS for the baseline hardware configuration). The two primary areas where supercomputing capabilities can have a major impact on the science return from the GLAST mission are (i) implementation of on-board pattern recognition and event analysis software that will provide the ability to analyze all gamma-ray and cosmic-ray events that trigger the instrument at the hardware level and, (ii) enable real-time analysis of transient events (e.g., the mysterious gamma-ray bursts) and autonomous response to these events. This response could take the form of requests for simultaneous data in real-time from other instruments (earth- or space-based) operating in the x-ray, optical, infrared, or microwave bands. In addition, onboard computing will have a central role in autonomously maintaining instrument calibration and determining alignment of the detector towers.

The availability of supercomputer capabilities in orbit would meet the baseline GLAST computing challenge and would extend the scientific reach of GLAST in important ways. In particular, supercomputing would allow implementation of more sophisticated on-board event triggering and processing that in turn would allow GLAST to (i) measure the energy spectra and elemental abundance of primary cosmic-rays up to some 10s of GeV and measure the flux and energy spectrum of electrons up to the TeV range, (ii) respond quickly to transient events such as high-energy gamma-ray bursts, and (iii) provide the additional computational capability needed to deal with the much larger event size (at least a factor of 5) associated with an imaging calorimeter. The imaging calorimeter can provide additional background rejection capability and enhance the gamma-ray astronomy reach of the instrument above 1 GeV by increasing the effective area at high energies by about a factor of 3 (therefore increasing the rate by a factor of 3 as well). Finally, on-board computing is necessary for monitoring the status of all instrument data channels, maintaining calibration, and determining the relative alignment of the silicon tracker planes. Relative alignment of the tracker channels needs to be known to about 50 µm. This can be accomplished in orbit by using high-energy cosmic-ray proton tracks (that provide straight tracks, relatively free of the effects of scattering) to internally survey the instrument. This alignment calibration will need to be done periodically throughout the mission. Establishing the ability to perform these functions effectively on-board can have important consequences for the actual design and operation of the GLAST. The availability of supercomputer capabilities will enhance instrument performance in all of these areas and has great potential for reducing ground operations cost by reducing the demand for high downlink capacity.

### Next Generation Space Telescope (NGST)

*Principal Investigator: Dr. John Mather (NASA Goddard Space Flight Center)*

In response to the recommendations of the *Hubble Space Telescope and Beyond Committee,* NASA is studying the feasibility of developing a large (8 meter diameter primary mirror) space telescope, optimized for use in the near infrared. The central mission for this instrument, dubbed the Next Generation Space Telescope (NGST), is the study of the early universe: the first stars and galactic structures that are thought to form at redshifts greater than those observable by the Hubble Space Telescope or other planned facilities. Supercomputer capabilities will have a major effect on the scientific capabilities of the NGST. The two primary areas for investigation are improvements in the data collection from large array detectors with 100 million pixels and improvements in control of the optical system. Improved data collection offers better sensitivity, better immunity to cosmic ray hits, and possibly better calibration accuracy, as well as a reduction in the amount of data to be sent to the ground. Better control of the optical system, which by its nature must be adjusted after launch, could yield better imaging and reduce the overhead of time spent adjusting the figure after it is disturbed. Progress in these areas would have major consequences for the actual design and operations of the NGST. The NGST study has defined a number of stretch technologies which could enable substantial improvements in scientific performance or reduction in cost. Onboard supercomputer capabilities fall in this category.

In the performance of multi-read infrared detector readout and signal processing, large gains in data compression and lowered noise appear possible but will require 100 - 1000 reads per pixel (up to 0.6 Gpixels per sec) and an algorithm to detect and eliminate cosmic rays. The NGST mission is baselined with a primitive version of such a program but larger gains appear possible leading to a reduction in requirements for down link bandwidth and onboard mass storage. With 100 million pixels, even a modest number of samples per second demands a very large compute capability, approaching GFLOPS or more. The computer memory needs to be large compared with the number of pixels, so at least 1 GB will be needed just for short term fast memory. We do not yet know whether a large memory will be required to hold a long time series for each image, with all 100-1000 reads in memory at once, or whether decisions can be made on the fly so that only a few samples per pixel are kept in the memory.

The availability of an on-board supercomputer will enhance the NGST mission optics in important ways. It will significantly increase the availability of the scientific instruments for scientific observations, by reducing the time required for the periodic fine-figure control. It will improve the quality of the imagery by allowing the adoption of potentially higher-performance closed-loop algorithms for fine-figure control. It would also make possible the adoption of much higher actuator-density deformable mirrors, such as are currently being developed at JPL for coronagraphic imagers. A coronagraphic camera with a second, 10,000 to 20,000 actuator deformable mirror will provide extremely high dynamic range imaging for direct planet detection.

### Mars Rover Science

*Principal Investigator: Dr. R. Stephen Saunders (NASA Jet Propulsion Laboratory)*

NASA has formulated a strategic framework for Mars exploration. The approach is to explore Mars along three thematic lines: search for life, understand climate history, and map resources and geology/geophysics. The strategy is to first obtain global geochemical and mineralogical maps of Mars from orbit. The second step is to characterize and explore sites using rovers that are capable of selecting samples of rock and soil. The third step is to land at one of the previously explored sites, collect a sample and return it to Earth. This strategy will be implemented in a series of missions that include a lander and orbiter in 2001 and in 2003 and the first sample return in 2005. The primary focus is on discovering whether life ever occurred on Mars, and if so, where and for how long. Future robotic missions to Mars, including missions with human crews who will work with robotic field assistants, will use supercomputer capabilities to greatly enhance the scientific return and capabilities of the next generation of Mars mobile platforms.

What is the new science we get with 100 times more computing power? We will develop a plan and partial implementation of software that will make use of 30 - 1000 MOPS/watt, in the range of 150 MOPS to 5 GOPS as compared to the Rover Sojourner at a few watts and perhaps 100 KOPS. The improvements fall into two categories. (1) Navigation: Basically, we want to get from point A to point B faster. The goal is a factor of 10 - 25 faster than Rocky 8 (The 2001 prototype), and access to at least 100 times more area during a mission. (2) Autonomous science operations. (2a) Autonomy to ensure science return in the event of missed commands. The total gain from autonomy is a factor of 6 - 9 in number of fast spectrometer measurements. (2b) Improved science along traverses. The additional science return from opportunistic autonomous observations along a traverse is a factor of about 50 over the return available without REE computing. When compared with the brute-force alternative of launching proportionately more missions to Mars, it is clear that REE computing will be enormously cost-effective for Mars Rover applications.

### Orbiting Thermal Imaging Spectrometer (OTIS)

*Principal Investigator: Prof. Alan R. Gillespie (University of Washington)*

NASA has currently deployed a thermal infrared spectrometer in orbit around Mars to determine surface components for which measurements of reflected sunlight are not diagnostic. Other governmental agencies are actively studying the role that thermal infrared imaging spectroscopy might play in remote sensing here on Earth, and they and NASA are now developing plans for hyperspectral thermal imagers in low Earth orbit. The purpose of these instruments will be not only to collect compositional information, but also to measure land surface temperatures with greater accuracy than has been possible before.

The key impediment to the accurate recovery of land-surface temperature and emissivity data is correction for atmospheric interference with the signal emitted from the land surface. Many approaches have been explored, and the most promising make use of in-scene measurements of the atmosphere rather than external data sources that lack spatial resolution, are taken at different times than the images, and don’t describe the boundary layer just above the surface. It is attractive to estimate atmospheric transmissivity and radiance, and to correct measured radiances for these parameters, at the point of data collection, in orbit.

The Sacagawea satellite proposed to NASA’s Earth System Science Pathfinder (ESSP) was based around a high-resolution HgCdTe imaging system that acquired 64 bands of thermal infrared radiance data at wavelengths from 8.3 to 11.6 µm, with a ground resolution of 30 m, an image swath width of 21 km and a temperature precision of 0.1K. Sacagawea also contained a separate imaging system to measure atmospheric effects at higher spectral resolution, but lower spatial resolution, in the wavelength region 7.5-8.5 µm, and a three-channel Visible/Near-Infrared (VNIR) imager to help distinguish vegetation, clouds and snow. Although Sacagawea itself will not be constructed under the ESSP program, a similar instrument is still under consideration.

On-board processing can be of great benefit in hyperspectral imaging to reduce data volumes and increase duty cycles. The focus of this application will be the development of an on-board processing system to (1) characterize and compensate for atmospheric effects, (2) calculate land surface temperatures and emissivity spectra, and (3) explore automated scene classifiers. In consideration of the diverse user community for these data, transmission of data to Earth may occur at different points in the processing stream. For data that have been completely processed to the thematic map level, data reduction by a factor of ~25 is feasible even before data compression. In extreme cases for which the scene is uniform (large forests, ice caps, water) greater savings are possible.

Atmospheric characterization will make use of a hybrid approach, using a combination of atmospheric data calculated from the atmospheric imager and estimated from forward models driven by climatological and topographic data, all augmented by empirical-line corrections over regions identified as having known surface types and emissivity spectra on the basis of the VNIR data.

### Solar Terrestrial Probe Multiplatform Missions

*Principal Investigator: Dr. Steven Curtis (NASA Goddard Space Flight Center)*

The Solar Terrestrial Probe line of missions was a result of the consensus on the direction of future missions across the Code S enterprise arrived at in Brekenridge, Colorado in 1997. The Solar Terrestrial Probe line is designed to be a series of scientifically linked to pursue a quantitative understanding of the flow of energy, momentum, and mass from the Sun, through interplanetary space, into the magnetosphere, and finally to where it is deposited in the Earth's upper atmosphere. The Solar Terrestrial Probe line is the logical successor to the highly successful International Solar Terrestrial Physics program which has provided the first system level study of the connections between the Sun and the Earth on global scales.

The proposed project will focus on multiplatform missions to study the Sun and the magnetosphere. These missions will consist of by 4 to 100 or more platforms flying in formation. The multiplatform requirement is driven by either image synthesis requirements for remote sensing, for example the low frequency radio imaging of solar processes or the need to uniquely separate space and time for in situ measurements on meso and micro scales, as is the case for the determination of electric currents from the curl of magnetic field variations. Each platform in these missions will have transmission to ground requirements in units of Gbits/day. Since there is an obvious burden on ground systems given the bandwidth requirements, a reduction in the amount of data transmitted to ground is necessary. This can be accomplished by onboard heuristic or high speed data analysis or a combination of both. The focus of this proposal is on the second path.

The tasks chosen for the proposed work are:

(1) plasma moment calculations for the constellation class nanospacecraft missions presently under study at GSFC as part of the Solar Terrestrial Probe line which are expected to fly in 2007 or later

**(2)** the calculation of cross correlations between pairs of time series for the imaging low frequency radio astronomy platforms which have been studied at GSFC under the Sun Earth Connections Mission New Concept program and earlier jointly with JPL under the similar Astrophysics New Mission Concepts Program

**(3)** the calculation of electrical current from magnetic field variations as measured by a cluster of four or more spacecraft as is being studied both for constellation class missions and for the Magnetospheric Multiscale, the latter of which is expected to fly in 2004 or later.